

1

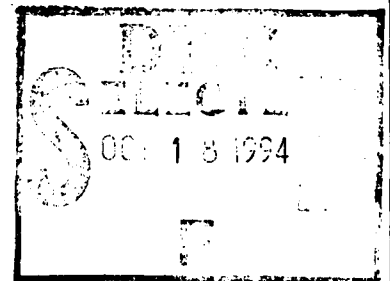
NAVAL POSTGRADUATE SCHOOL

Monterey, California

AD-A285 525



THESIS



SECURE DISTRIBUTED FILE SYSTEMS

by

Tracy Michael Conroy
and
Winslow Hurlburt Buxton

September, 1994

Thesis Advisor:

Roger Stemp

Approved for public release; distribution is unlimited.

DTIC QUALITY INSPECTED 8

94-32360



11898

9410

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704	
<p>Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.</p>				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE September 1994		3. REPORT TYPE AND DATES COVERED Master's Thesis
4. TITLE AND SUBTITLE SECURE DISTRIBUTED FILE SYSTEMS UNCLASSIFIED			5. FUNDING NUMBERS	
6. AUTHOR(S) Winslow H. Buxton, Tracy M. Conroy				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey CA 93943-5000			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)			10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.			12b. DISTRIBUTION CODE *A	
13. ABSTRACT (maximum 200 words) Secure information distribution is a strategic capability as significant as weapons systems and tactics to military operations. The Department of Defense has recognized the importance of establishing and maintaining secure distributed communications between automated information systems. This research reviews eleven different distributed file systems and explores the practicality and applicability of one such system, Trusted Ficus File System (TRUFFLES), in the DoD infrastructure. Integrated into this research are discussions of Privacy Enhanced Mail (PEM), which is currently an integral part of the TRUFFLES implementation. This thesis concludes with a discussion of the actual installation of a PEM reference implementation, and future requirements for the TRUFFLES installation at the Naval Postgraduate School.				
14. Secure Distributed File System, Replication, Digital Signatures, Encryption, Digital Encryption Standard (DES), Privacy Enhanced Mail (PEM)			15. NUMBER OF PAGES 118	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL	

Approved for public release; distribution is unlimited.

Secure Distributed File Systems

by

Tracy M. Conroy
Lieutenant Commander, United States Navy
B.A., Rutgers University, 1982


Winslow H. Buxton
Lieutenant, United States Navy
B.S., University of Oregon, 1983

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN
INFORMATION TECHNOLOGY MANAGEMENT

from the
NAVAL POSTGRADUATE SCHOOL
September 1994

Author:



Tracy M. Conroy

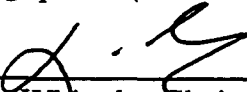
Author:


Winslow H. Buxton

Approved by:


Roger Stemp, Principal Advisor


Kishore Sengupta, Associate Advisor


David R. Whipple, Chairman
Department of Systems Management

ABSTRACT

Secure information distribution is a strategic capability as significant as weapons systems and tactics to military operations. The Department of Defense has recognized the importance of establishing and maintaining secure distributed communications between automated information systems. This research reviews eleven different distributed file systems and explores the practicality and applicability of one such system, Trusted Ficus File System (TRUFFLES), in the DoD infrastructure. Integrated into this research are discussions of Privacy Enhanced Mail (PEM), which is currently an integral part of the TRUFFLES implementation. This thesis concludes with a discussion of the actual installation of a PEM reference implementation, and future requirements for the TRUFFLES installation at the Naval Postgraduate School.

Accession For	
NTIS	<input checked="checked" type="checkbox"/>
CRA&I	<input type="checkbox"/>
DTIC	<input type="checkbox"/>
TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail and/or Special
A-1	

TABLE OF CONTENTS

I.	INTRODUCTION	1
A.	THE BASIC RESEARCH QUESTION	3
B.	DISCUSSION OF THE BASIC RESEARCH QUESTIONS	4
II.	BACKGROUND INFORMATION	5
A.	DISTRIBUTED FILE SYSTEMS	5
B.	SECURITY AND SYSTEM ARCHITECTURES	6
	1. Definition of Threats and Vulnerabilities; Security Versus Ease of Use	9
	2. Incorporation Into the Existing Network Structure; GOSIP, X.400, X.500, X.509	11
C.	ASYMMETRIC AND SYMMETRIC CRYPTOGRAPHY	13
D.	DIGITAL SIGNATURES AND ENCRYPTION ALGORITHMS	15
E.	PRIVACY ENHANCED MAIL	20
	1. Trusted Information System's-Privacy Enhanced Mail (TIS/PEM)	23
	2. Riordan's Internet Privacy Enhanced Mail (RIPEM)	24
F.	SYSTEM ADMINISTRATION AND KEY MANAGEMENT ISSUES	26
G.	PASSWORD/LOGIN SYSTEMS	29
H.	FIREWALL SYSTEMS	31
I.	ISSUES ON AUTHENTICITY OF DATA AND USERS	35

III.	DISTRIBUTED SYSTEMS AND SERVICES	38
A.	SUN NETWORK FILE SYSTEM (NFS)	39
1.	System Design Goals	39
2.	Theory of Operation	40
3.	Authentication, Security and File Sharing Issues	42
B.	THE ANDREW FILE SYSTEM (AFS)	43
1.	System Design Goals	43
2.	Theory of Operation	44
3.	Authentication, Security and File Sharing Issues	46
C.	THE CODA FILE SYSTEM	48
1.	System Design Goals	48
2.	Theory of Operation	49
3.	Authentication, Security and File Sharing Issues	53
D.	APOLLO DOMAIN FILE SYSTEM	54
1.	System Design Goals	54
2.	Theory of Operation	55
3.	Authentication, Security and File Sharing Issues	56
E.	AT&T REMOTE FILE SHARING	57
1.	System Design Goals	57
2.	Theory of Operation	58
3.	Authentication, Security and File Sharing Issues	59

F.	SPRITE NETWORK FILE SYSTEM	60
1.	System Design Goals	60
2.	Theory of Operation	60
3.	Authentication, Security and File Sharing Issues	61
G.	IBM AIX DISTRIBUTED SERVICES	62
1.	System Design Goals	62
2.	Theory of Operation	62
3.	Authentication, Security and File Sharing Issues	63
H.	THE LOCUS DISTRIBUTED OPERATING SYSTEM	65
1.	System Design Goals	65
2.	Theory of Operation	66
3.	Authentication, Security and File Sharing Issues	67
I.	PROJECT ATHENA AND KERBEROS	69
1.	System Design Goals	69
2.	Theory of Operation	70
3.	Authentication, Security and File Sharing Issues	71
J.	THE FICUS DISTRIBUTED FILE SYSTEM	72
1.	System Design Goals	72
2.	Theory of Operation	73
3.	Authentication, Security and File Sharing Issues	77

K. COMMON CHARACTERISTICS OF REVIEWED DISTRIBUTED FILE SYSTEMS	77
IV. TRUFFLES	80
A. DISTRIBUTED FILE SYSTEMS AND TRUFFLES	80
B. TRUFFLES	82
C. FILES/VOLUMES	84
D. TRUFFLES ARCHITECTURE	86
E. TIS/PEM ARCHITECTURE	87
F. PRACTICALITY OF INSTALLING TRUFFLES AT NPS	88
G. TIS/PEM INSTALLATION AT NPS	89
V. CONCLUSIONS	93
APPENDIX A: THE OSI MODEL	99
APPENDIX B: ENCRYPTED PEM MESSAGE	102
APPENDIX C: ACQUIRING TIS/PEM	104
REFERENCES	106
INITIAL DISTRIBUTION LIST	109

LIST OF FIGURES

Figure 1. Government OSI Protocols (GOSIP)	11
Figure 2. SUN Network File System	41
Figure 3. Andrew File System (AFS)	44
Figure 4. Overall View of Coda	51
Figure 5. Coda in Connected and Disconnected Modes . .	52
Figure 6. Apollo Domain File System	56
Figure 7. AT&T Remote File Sharing	58
Figure 8. Sprite Network File System	61
Figure 9. IBM AIX Distributed Services	64
Figure 10. The LOCUS File System	66
Figure 11. Kerberos Authentication Scheme	70
Figure 12. Logical Ficus Volume Set or Container . . .	74
Figure 13. Ficus File Modifications	75
Figure 14. Ficus Stackable Layer Interface	76
Figure 15. File Hierarchies Using Truffles	84
Figure 16. TRUFFLES Architecture	86
Figure 17. TIS/PEM Interface Architecture	87
Figure 18. OSI Model	100

I. INTRODUCTION

In this chapter the issues discussed are the basic research question, application of this thesis to the Department of Defense (DoD), and a discussion of the need for a secure electronic large volume file transfer system within the DoD.

The United States Military is becoming increasingly dependent on the inter-connection and communications of automated information systems. In 1989, the DoD spent more than nine billion dollars in general purpose automated data processing equipment, software and related services. This information technology budget represents a commitment by the DoD to "tens of billions" in future expenditures.¹

Information is a strategic resource as significant as people and material to military commanders. Systems used by the military impart specific knowledge which lead to informed decisions; and informed decisions lead to victories.²

The advances in computing will further increase federal demand for information technology products and services. One

¹Committee of Government Operations, Sixth Report on DoD Automated Information Systems, Report 101-382, November 16, 1989.

²Kellner, Mark, "Data Power", Government Executive, August 1991.

source of growth is the increased use of open computer systems based on the UNIX³ standard.⁴

On November 16, 1990, the Secretary of Defense directed the implementation of the DoD Corporate Information Management (CIM) initiative. Some of the objectives of the CIM initiative are to improve portability, inter-operability, vendor independence and security of DoD information systems and architectures. A key element to the CIM initiative is in the implementation of wide area computing and development of a communications infrastructure which will support these objectives.

With the rise in federal spending on computer systems and inter-connection, clearly an increased emphasis is being placed on information security classification, distributed communications, file sharing and replication systems, and in the infrastructure necessary for the secure communication of information shared between systems. Large amounts of textual and non-textual data must be transferred in a secure manner and with the same "user ease" of logging onto and transferring information through a typical communications network electronic mail system.

³UNIX is the operating system developed and trademarked by AT&T. UNIX is a pun on the Multics operating system, developed by MIT in the 1960s.

⁴Kellner, M., "Data Power", Government Executive, August 1991.

Currently, there are several systems under development for ensuring the secure transport of data, the authentication of the users, and user file sharing and replication. These systems generally fall into one of two categories: application or software file structures which utilizes cryptology for encoding the data prior to transmission; and application software/hardware "Firewall" type systems which are separated from the message and act as a "sentry" into or out of an individually administered computer network system.

A. THE BASIC RESEARCH QUESTION

This research will review different distributed file systems and explore the practicality and applicability of one such system in a DoD infrastructure. The following topics will further explain and amplify this research question.

1. What is a Secure Distributed File System?
2. What are the qualities and technologies a Distributed File System should have for incorporation into the structure of the Department of Defense?
3. What types of systems are currently available and how do they compare in the following areas: System design goals, theory of operation and general authentication, security, and file sharing issues?
4. What secure distributed file system currently available can best fit the needs of the DoD? How can it be implemented into an existing network system; more specifically, implemented into an existing network system at the Naval Postgraduate School.

B. DISCUSSION OF THE BASIC RESEARCH QUESTIONS

The research will also look at the definitions and terms used, encryption methodologies for authentication, and security requirements of a secure electronic distributed file transfer system over common (untrusted) lines. Systems considered are the Sun Network File System (NFS), the Andrew file system, the Coda File System, the Apollo Domain File System (ADFS), the AT&T Remote File Sharing System (RFS), the Sprite Network File System (SNFS), IBM AIX Distributed Services, the Locus Operating System, Project Athena and Kerberos, the Ficus File System and the Trusted Files Ficus System (TRUFFLES). The paper concludes with a discussion of the selection of one of these systems, TRUFFLES, for incorporation into a DoD infrastructure. Also discussed are issues arising from the actual implementation of Privacy Enhanced Mail and further implementation requirements for TRUFFLES at the Naval Postgraduate School.

II. BACKGROUND INFORMATION

Secure electronic large volume distributed file transfer systems currently under development incorporate a wide means of ensuring data security and authenticity. This chapter discusses the first two research questions; what is a secure distributed file system, and what are the qualities and technologies a distributed system should have for incorporation into the structure of the DoD. Background information is presented here for reference into current trends and procedures in the development of the systems.

A. DISTRIBUTED FILE SYSTEMS

In general, a file system is composed of a portion of the computers operating system whose function is to store or retrieve data from a storage medium, usually a disk or tape drive. A distributed file system allows for the sharing or replication of electronic files over a network. A secure distributed file system incorporates an end to end encryption scheme for the data while in the transfer mode.

A distributed system may be viewed as a set of logically related functional components and not as a set of connected computer systems. It provides for reliability from the use of multiple components located in many areas; efficiency in data

access and overall system response time; flexibility in the incremental upscaling of computing power by the utilization of coordinating moderate sized computer systems.⁵

Typically, distributed file systems functionality resides in the application or presentation layer of the OSI seven layer protocol reference model.⁶ A protocol is a formal description of a message format and the communication rules the connected computers must follow in order to exchange data. They describe all details of the machine interfaces from the lower or physical level to the top or application level.

Within the Internet, a majority of data access and transfer is through File Transfer Protocol (FTP). In a more tightly configured integrated computing environment, Sun's Network File System (NFS) is predominantly used.⁷

B. SECURITY AND SYSTEM ARCHITECTURES

This section discusses issues concerning secure information transfer over wide area networks, Discretionary Access Controls (DAC's) for such a network and DoD specific standards on levels of information security.

⁵Muftic, Sead, "Security Mechanisms for Computer Networks", Ellis Horwood Limited, 1989, pg 23, 159.

⁶See APPENDIX A.

⁷Malamud, C., 'STACKS, Interoperability in Today's Computer Networks", Prentice Hall, 1992, pg 11-16.

The Department of Defense Standard Trusted Computer System Evaluation Criteria (DoD 5200.28-STD) states that: "Any discussion of computer security necessarily starts with a statement of requirements. In general, secure systems will control access to information such that only properly authorized individuals will have access to read, write, create or delete information."⁶ It further clarifies that there are six fundamental requirements derived from this basic statement: four of which provide control to information accesses and two which provide creditable information assurances.

Information security standards are divided into four divisions: A, B, C, and D. Division A is applied to systems with the most comprehensive level of security. Division D is considered the lowest security level; almost any computer system begins at the lowest level (no security assurance) and assumes higher divisional ratings as hardware or software components are incorporated. This provides a method for measuring an increased level of confidence for the protection of more sensitive information. Within divisions B and C are class subdivisions which are also ranked in a hierarchical order. These classifications generally pertain to hardware and software systems and not to the information possessed on the

⁶DoD Trusted computer system Evaluation criteria, DoD 5200.28-std, Library No. S225,711, December 1985.

machine. General security information divisions (data) are broken into the following levels:

- Level 1 data for classified or secure information.
- Level 2 data for information which is unclassified but considered sensitive.
- Level 3 data for unclassified and non-sensitive information.

Discretionary Access Controls (DAC's) are a method of restricting access to files based on the identity of the user and are the most common control methods used in trusted systems.⁹

A majority of the information sent via electronic means are of data security levels 2 or 3. Requirement guidelines for this level of information suggests an Orange Book security level of C2-B1¹⁰ and are in specific realms of:

- Identification and Authentication: establishing the identity of the user and verifying that the user is who he/she claims to be.
- Access Control: allowing users to access only those resources to which they are authorized; and protect against unauthorized access.
- Audit Trail: ensuring that sufficient information is recorded of each event for non-repudiation of a security

⁹Russell, D. and Gangemi, G.T., "Computer Security Basics", O'Reilly and Associates, Inc, July 1992.

¹⁰The exact system requirements are unknown, but are assumed to be similar in scope to the security classification of the software system as delineated in the document "B3 or F-B3 Security Target (DRAFT)" by Trusted Information Systems, Inc; Document No: TIS TMACH Edoc-0005-93B; April 30, 1993.

investigation by a third party user, a difference between C & B.

- Trusted Path: ensure a trusted communications path between the user and the computing base. It is highly resistant to penetration.
- Security Domain: a security administrator is supported and mechanisms are utilized to signal and correct security related events with established recovery procedures.

1. Definition of Threats and Vulnerabilities; Security Versus Ease of Use

With the increasing demands on timely information and the greater volumes of information being processed, data protection and replication becomes a major security issues. Comprehensive information resource protection procedures must address, at a minimum, the accountability of the information and the users involved, data access and hardware/software controls. The greater the value of information processed, the greater the consequences for its unauthorized use; and therefore, the greater the need of control measures to protected it.¹¹

An assumption often made about threats and vulnerabilities are that they are separate and distinct entities, or that they are attached to a sole technical

¹¹Helsing, Cheryl, "Executive Guide to the Protection of Information Resources", National Institute of Standards and Technology, date unknown.

process.¹² For the purpose of this paper, a threat is anything which can be perceived as a possible danger to the information system. Threats can be broken down into two categories: Passive and Active.

A Passive threat involves the interception but not the alteration of information. The danger of the passive threat is the unknown compromise of sensitive information.

An Active threat involves the interception and alteration of information. An active threat's primary danger is in the falsified authenticity of the information.

In contrast, a vulnerability is a weakness in a computer system where it is susceptible to attack and can be deliberately exploited to violate system security. A threat to an information system will exploit the vulnerability of that system, violating the integrity and security of that system.

In general, the more secure a system, the less "user friendly" that system becomes. While security measures are important, procedures or devices which are not practical or cumbersome are often circumvented by legitimate users in order to complete an assigned job. Because of this humanistic

¹²Koerner, Frank, "System Threats and Vulnerabilities and the Contrary Principle", Computer Humanware International, Elsevier Science Publishers Ltd, 1993.

feature, usability or "user friendliness" becomes an extremely important but often overlooked security feature.¹³

2. Incorporation Into the Existing Network Structure; GOSIP, X.400, X.500, X.509

Starting in 1990, Government agencies which required computer networking products were required to buy systems which conformed to a standard set of Government OSI protocols (GOSIP). This collection of protocols was sponsored by the National Institute of Standards Technology (NIST) and is the United States government's subset version of international OSI standards. In the United States the GOSIP profile is as follows:

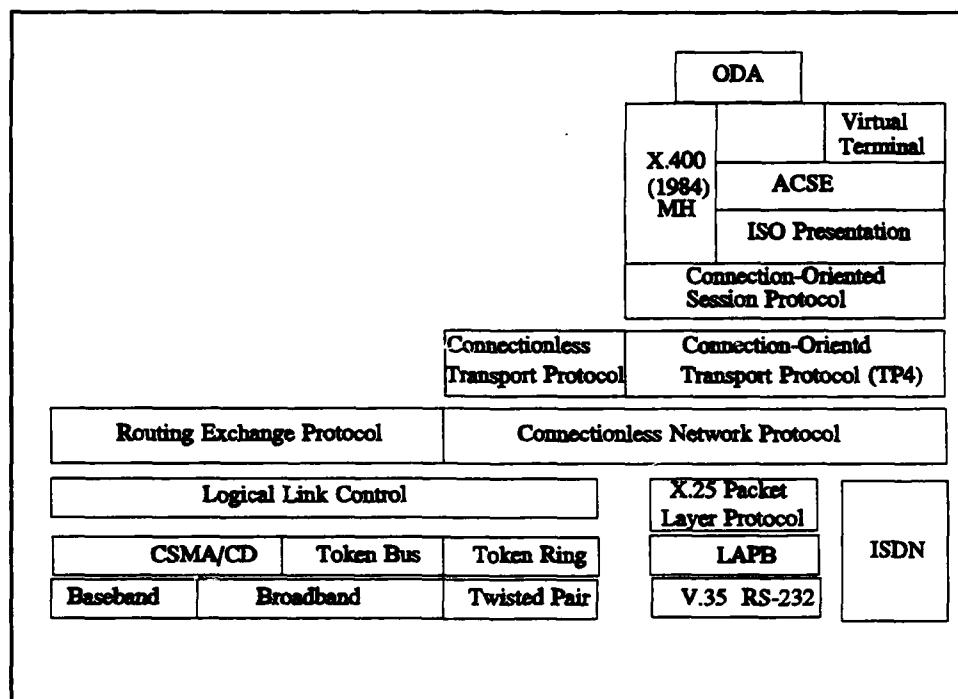


Figure 1. Government OSI Protocols (GOSIP)

¹³Avolio, F.M., Ranum, M.J., "A Network Perimeter with Secure External Access", Trusted Information Systems, Inc, January 25, 1994.

It requires the File Transfer, Access and Management protocol¹⁴(FTAM) for file access services and X.400 protocol for messaging services.

The FTAM protocol is considered systematically complex and provides a unified structure for the following areas: virtual filestore definitions, file service definitions and supporting file protocol specifications.¹⁵

The virtual filestore definition portion of FTAM provides for a comprehensive structured definition of file accesses, presentations, data transfer and identification. The file service definition portion allows for a structured definition of services available to users and establishes user authority or privileges to access and manipulate files. The file protocol specifications of FTAM provide direct support by assigning "one to one" mapping of service applications to specific data points. The protocol establishes a session connection and allows for the flow of data through specified "checkpoints".

X.400 is an International Consultative Committee for Telegraphy and Telephony (CCITT) standard and an International Standards Organization (ISO) protocol for message handling systems and is being accepted as a standard for store and forward message delivery. In UNIX, message handling is done by

¹⁴See APPENDIX A.

¹⁵Stallings, W., Data and Computer Communications, 4th Edition, MacMillian Publishing Company, New York, pgs 729-735.

UNIX-to-UNIX Copy Program (UUCP)¹⁶ and in Transmission Control Protocol/Internet Protocol(TCP/IP) by Simple Mail Transfer Protocol (SMTP). X.400 is the protocol which connects all these systems together into one general message handling system. Most vendors have X.400 gateways in their systems which translate addresses and message formats into an appropriate format for the target systems. As a result, gateway systems are becoming widespread.¹⁷

X.500 is a CCITT standard protocol for defining directory information and X.509 is a CCITT standard directory for establishing an authentication framework. X.500 is generally used in conjunction with X.400 and provides for a global directory for use in OSI networks. This allows for naming services and name searches and is based on a strict hierarchy which assumes each administrative domain provides information on its members.

C. ASYMMETRIC AND SYMMETRIC CRYPTOGRAPHY

Generally, two types of cryptographic systems are used for network security: one key or symmetric systems and public/private key or asymmetric systems. In symmetric cryptography, the enciphering and deciphering keys are usually

¹⁶UUCP is the standard UNIX utility used to exchange information between any two UNIX nodes.

¹⁷Malamud, Carl, "STACKS: Interoperability in Today's Computer Networks", Prentice Hall, Englewood Cliffs, New Jersey, 1992.

the same and must be transmitted from the sender to the receiver via some form of secure means to ensure confidentiality and integrity.

The Data Encryption Standard (DES) is the algorithm most utilized in symmetric cryptography in the United States. It uses a key length of 56 bits and has been recommended for use by non-military government agencies. The DES algorithm has been considered effective. A small change in the plain text message produces a large change in the cipher text (known as the avalanche effect) and exhibits a strong complimentary property between the plain text, cipher text, and key. However, it has recently come under some criticism as to the choice of the 56 bit key length which is considered by many as being too small. The advancement of computer technology in producing increasingly sophisticated microprocessors at a lower cost legitimizes the claim that the key could be deciphered by either a special purpose machine or via "brute force" using a table look up approach.¹⁸

In asymmetric cryptography the enciphering and deciphering keys are related mathematically but are considered to be computationally infeasible to decipher one from the other; therefore, one key (used for enciphering) can be made public and the deciphering key is kept private. This type of system avoids the necessity of transmitting a secure key over an

¹⁸Muftic, Sead, "Security Mechanisms for Computer Networks", Ellis Horwood Limited, 1989, Sec 2.2 pg 50.

untrusted medium and can be used to transmit a secure key of a symmetric system over a non-secure line.

The concept of asymmetric cryptography is the following: "Bob" encrypts a message using the public key of "Alice" and then sends the message to Alice over an untrusted channel. Only Alice can decrypt the message since she is the only one who holds her private key.¹⁹

In the above example, if Bob were to "sign" the message with his private key then Alice could recognize Bob's signature with his public key. In this situation, asymmetrical systems also allow for the utilization of digital signatures. The algorithm used in most public/private key crypto-systems is the RSA or Rivest Shamir Adleman algorithm.

D. DIGITAL SIGNATURES AND ENCRYPTION ALGORITHMS

A Digital Signature is a bit string attached to an electronic document which is generated by a signee and is based on both the document's data and the individual's secret password. Anyone who receives the document can legally prove the signee of the document (non-repudiation) and whether it was altered.²⁰

¹⁹ $H_m = E_{\text{private}}(M)$; once message is sent and received then $E_{\text{public}}(M) = H_m$.

²⁰Schneier, Bruce, "Digital Signatures", BYTE magazine, November 1993, pg 309.

A digital signature is useful for many reasons. First, it is not forgeable and serves as proof to the recipient that the signer deliberately signed the electronic document. Second, the signature is not reusable and is part of the document. It cannot be transferred to a different document. Finally, once signed, the document is unalterable and the signature cannot be repudiated.

Digital signatures are used to detect unauthorized modifications to data and to authenticate the identity of the signatory. In addition, the recipient can use the digital signature to prove to a third party that the signature was generated by the signatory in a method known as non-repudiation.²¹

Asymmetric or public key cryptography is used for digital signatures. The NIST standard cryptological method is the Digital Signature Algorithm (DSA). Public Key cryptography uses the DSA encryption with two different keys, the public keys, which are known to the public in general, and private keys which are known only to the individual signing the message. DSA will encrypt the message or file using both keys; however, signature generation can only be performed by use of the private key.

A hash function is used in the signature generation process to obtain a condensed version of the data, called, the

²¹Digital Signature Standard (DRAFT), National Institute of Standards and Technology, February 1, 1993.

message digest. Along with the keys, the message digest is used as an input into the DSA to generate the digital signature.

The recipient of the message, using the public key, will employ the same hash function to decrypt the message and verify the signature. If the file or message has been altered in any way, the hash function will fail and the document will be undecipherable.

In practical applications, DSA is too inefficient to sign large documents directly. To account for this, DSA is implemented with a Standard Hash Algorithm (SHA) which was designed by NIST for use with DSA. The result of this is to allow the signatory to sign the hash of the document rather than the document itself.

Another encryption algorithm, the RSA algorithm, developed by RSA Data Security, can be used to implement digital signatures. The algorithm was created by Ron Rivest, Adi Shamir and Leonard Adelman and became an ISO standard but was not used by NIST because of patent problems. RSA is patented in this country but not abroad.²²

Along with a difference in algorithm structure, RSA and DSA also differ in key size. DSA originally used 512 bit keys which were thought by some to be too short for long term security projects or special contract agreements which must

²²Schneier, Bruce, "Digital Signatures", Byte magazine, November 1993, pg 312.

remain intact for several years. Using a specially designed computer with parallel processing, a 512 bit key could theoretically be cracked in a matter of months. To counter this NIST made the keys variable between 512 and 1024 bits, which would take several hundred years to crack with existing technology.²³

The security of a digital signature is also dependent on the maintenance of secrecy of the users private key. For this reason, conformance to this standard and the use of a digital signature does not insure that a particular implementation is secure, but only that the document is secure with reference to the signature applied to it. The responsible authority in each agency must ensure the that implementation of digital signatures provides for an acceptable level of security, and that there must be a binding of the user's identity and the user's public key.

When a signed message is directly transmitted from a sender to a receiver without any arbitrator, it is called a true digital signature. However, if the sender wants to refute the true signature, he can make his private encoding signature public and claim it was a forgery. To prevent this, an arbitrator is used which results in an arbitrated digital signature. With this method, the sender and the arbitrator use a common key to encode the message and create the digital

²³ibid, pg 312.

signature. If a conflict arises, then the arbitrator can verify the authenticity of the signature and legally remove all doubt as to its authorship.

Another cryptographic device under consideration is the Clipper chip, under development and intended for use by the federal government. The Clipper chip is a cryptographic device intended to protect private computer communications while still allowing specific federal government law enforcement agencies access to the enciphered communications through the use of governmentally held keys. Clipper is intended for use in encrypting voice transmission while a similar device, the Capstone chip, will be used to encrypt data communications.

The encoding algorithm used in Clipper and Capstone is the Skipjack algorithm which was developed by the National Security Agency (NSA). It uses 80 bit keys, with 32 rounds of scrambling (DES uses 56 bit keys and 16 rounds of scrambling) and supports all DES operating modes.²⁴

The Skipjack algorithm was evaluated in July, 1993, by the following individuals: E. Brickell of Sandia National Laboratories, D. Denning of Georgetown University, S. Kent of BBN Communications Corp., D. Maher of AT&T, and W. Tuchman of Amperif Corp. The results of their evaluations were as follows:

²⁴Denning, D., The Clipper Chip: A Technical Summary, April, 1993.

1. It will be 36 years before the cost of an exhaustive search will be equal to the cost of breaking DES today. There is no significant risk that Skipjack will be broken by an exhaustive search in the next 30 to 40 years.
2. There is no significant risk that Skipjack can be broken through a shortcut method of attack.
3. The internal structure of Skipjack must be classified for policy protection, but its strength against an attack does not depend on the secrecy of the algorithm.

E. PRIVACY ENHANCED MAIL

Privacy Enhanced Mail (PEM) is an Internet standard which allows for message and sender authentication and confidentiality through electronic mail. It uses cryptographic techniques to provide for message integrity, originator authentication and confidentiality. All privacy enhancements are implemented at the application layer and are not dependant on any other privacy enhanced feature at a lower protocol level.

PEM uses X.509 certificates to attach a distinguished name to an RSA or DSA Public Key. A distinguished name is a name which uniquely globally identifies a user. A certification authority vouches for the binding of a distinguished name and the public key associated with it in the organizational unit to which the name is attached. The distinguished name starts with the organization's name and then attaches some character string which identifies the user, generally a common name.

PEM services are offered through the use of end to end cryptography at or above the user agent level. No special processing requirements are imposed on the Message Transfer Service at the end points or at intermediate relay sites. This allows for PEM facilities to be incorporated selectively on a site by site or user basis without impacting other Internet services.²⁵ However, it is necessary for the sender to know whether the intended recipient uses PEM, otherwise the message will be encoded and the recipient will not be able to provide the inverse transformation needed to decode it.

Obviously, key management becomes a very important issue. The concept of public key certificates are defined in the X.509 architecture. A public key certificate contains the name of the user, his public key, and the name of the issuer which vouches that the public key is bonded to the user. This data, along with a valid time interval, is cryptographically signed via a digital signature by the user's private key. The user and issuer names are distinguished names as defined in X.500 architecture and discussed earlier. Once a message is digitally signed, it can be stored or transmitted via non-secure lines.²⁶

²⁵Linn, J. "Privacy Enhancement for Internet Electronic Mail: Part I Message Encryption and Authentication Procedures", Network Working Group RFC 1421, February 1993.

²⁶Kent, S. "Privacy Enhancement for Internet Electronic Mail: Part II Certificate-based Key Management", Network Working Group RFC 1422, February 1993.

Prior to sending an encrypted message, the sender must acquire the certificate for each recipient and validate them by checking the digital signature associated with each. Once validated, the public key is taken from the certificate and is used to encrypt the unique Data Encryption Key (DEK) which is used to encrypt the message. Upon receipt of the message, the recipient uses their own mathematically related private key to decrypt the DEK which is then used to decrypt the message.

Keying of PEM transmissions uses a two level key hierarchy. First are the Data Encryption Keys which are used for encryption of the message body. In asymmetrical key techniques, the DEKs are also used to encrypt the signed representation, or hash, of the message. A DEK is unique to the message transmitted and therefore no pre-distribution of DEKs are required.

The second key needed is the Interchange Key (IKs) which is used to encrypt the DEKs for the transmission of the message. The same IK will be used by a given originator to a known recipient. Given the correct IK, the recipient can decrypt the transmitted DEK, which in turn is used to decrypt the message body (See APPENDIX B for sample PEM message).

If symmetric cryptography is used for the DEK transmission, then the IK is a single key used by both the sender and recipient. This key will encrypt both the DEK and the message body. If asymmetrical cryptography is used, the IK is the public key of the recipient and is used to encrypt

the DEK. The IK used to encrypt the message body is the private key of the sender.

Currently there are two implementations of PEM available, TIS/PEM and RIPEM.

1. Trusted Information System's-Privacy Enhanced Mail (TIS/PEM)

Trusted Information Systems Inc. (TIS), in cooperation with RSA Data Security Inc. and under an ARPA sponsorship have developed a UNIX based implementation of privacy enhanced mail called TIS/PEM.

TIS/PEM authentication is done by RSA public key cryptography and uses the Data Encryption Standard (DES). The message is encrypted using the DES key, which is in turn encrypted by the public key of the receiver and a hash of the message encrypted by the private key of the sender. The receiver uses his private key to decrypt the DES key, which is then used to decrypt the message. It compares this message to the hash generated by applying the sender's public key to the encrypted hash. Every message uses a new DES key, and users maintain their own public and private keys.²⁷

TIS/PEM uses X.509 certificates to bind an X.500 distinguished name to a public key, which is vouched for by a TIS managed Certification Authority. It is a reference

²⁷P. Reiher, T. Page, G. Popek, J. Cook, "TRUFFLES - A Secure Service for Widespread File Sharing", Trusted Information Systems, Inc. 1992.

implementation of the PEM standard, is UNIX based, and runs on a variety of platforms. A PEM library serves as a primary entry point into the system. Key management administrators and programs communicate with a local key manager who coordinates local key management independent of the particular application requesting its services. The key manager maintains a local database for the certificates and private keys, enforces access control, and provides for cryptographic services employing private keys.

One of the libraries attached to TIS/PEM is a cryptological library which has an algorithm independent interface and handles key generation, message digest computation, encryption and decryption, and signature computation and verification.

TIS/PEM is distributed in source form and is available within the US and Canada for non-commercial use. It cannot be exported.

2. Riordan's Internet Privacy Enhanced Mail (RIPEM)

Riordan's Internet Privacy Enhanced Mail (RIPEM) is a public key encryption program for use with electronic mail. It allows an individual to generate public keys and encrypt/decrypt using those keys. It uses conventional asymmetrical cryptography and provides capabilities similar to PEM.

RIPEM uses the concept of a certificate to bind a key to a user, but does not employ the full certificate hierarchy as used in other PEM systems. Instead, it uses a "Direct Trust" model where users certify each other without the use of a third party corporate hierarchy.²⁸

RIPEM generates a pseudo-random message key which it uses to encode the message using a symmetric key encryption algorithm. The message key is then encoded using the RSA public key algorithm and includes the encoded message key within the message. The advantage of this method is that the DES encryption used in this manner is much faster than in the other public key systems.²⁹

The signature of the message is computed by either a "checksum" or hash of the message plaintext, and then encrypting this hash with the sender's private key and decoded with the sender's public key, which is the reverse of other PEM methods. Rivests' MD5 message digest algorithm, instead of the Standard Hash Algorithm, is used for the hash function.

After the recipient verifies the sender's signature, he computes his own message digest of the message after decrypting it. The recipient then decrypts the encrypted message digest using the sender's public key and checks it

²⁸Riordan, M., "RIPEM Users Guide: for RIPEM vers. 1.2", January 1994.

²⁹ibid.

against the recomputed digest. If they both match, then the sender is verified.

RIPEM will run on MS-DOS, Macintosh, OS/2, Windows NT and UNIX operating systems.

F. SYSTEM ADMINISTRATION AND KEY MANAGEMENT ISSUES

Both types of cryptography systems require the use of proper key management techniques. In general, key generation should not be user selected but rather a randomly generated selection process. Private keys must never be stored on any medium which can be copied or read and should be stored in an encrypted form.

There are two principle methods of key distribution. The first is the use of Key Distribution Centers (KDC). In a KDC all keys emanate from a single source which keeps track of key origination and distribution. It controls which keys are distributed to whom and control all messages within their administration. The advantage of the central administration of key distribution is to prevent issuance of duplicate or weak keys. The disadvantage is that corruption of the distribution center has extensive repercussions on the security of all sessions in the system administered by that center.

The second method of key distribution is through direct exchange of a session key by communicating partners. The advantage is the limited distribution of keys in controlling

the access to each session which protects other sessions should the keys be compromised. The disadvantage lies in the coordination required to establish a reliable mutual authenticity of the communicating partners.

The problem associated with both of these methods is identified in a concept called the "Conversation Reality"³⁰ in which the keys are exchanged in such a manner to prevent replay or immediate reuse of the same key. There are currently two methods known to help solve this problem.

The first is a Challenge - Response system: User A wants to ensure user B's message is not a replay, he includes an unpredictable element or challenge in the message to B. B responds by using some standard function in response that could not have been prepared in advance. After receiving and confirming B's response, A is sure that the conversation with B is not a replay.

The second is a timestamp mechanism in which each message is given a timestamp for survival. The user can check the age of the message and decide for themselves if the message is too old and possibly a replay.

In both of these cases, encryption of the challenge-response or timestamp should be used to ensure that an intruder did not construct the response or modify the timestamp. Therefore, the problem of key management is reduced

³⁰Muftic, Sead, "Security Mechanisms for Computer Networks", Ellis Horwood Limited, 1989, pg 66.

to finding a key distribution protocol with the following properties:³¹

1. No need for a Central Server
2. Mutual authentication of communicating parties
3. A verifiable conversation by a challenge response system
4. A minimal number of messages to exchange the keys.

Key exchange can either be through a direct exchange, or an exchange via a third party. In each of the methods, it is necessary that each user has a direct authenticator. A direct authenticator may be in or outside of the network but must be known to each of the users to authenticate the session. In a direct exchange, the users may be able to authenticate each other. In a third party exchange, the authenticator may be another (trusted) person who authenticates each user individually if the two users cannot authenticate each other directly.

In current architecture, X.509 provides a framework for authentication of entities involved in a distributed directory service by the use of public key crypto-systems.

A certificate is central to key management in the X.509 architecture and PEM. The process for authenticating both parties in message transmission should be simple, automated and uniform. Although certificate management systems are

³¹Ibid, pg 67.

compatible with different digital signature algorithms, RSA is becoming the established central crypto-system and has greater use as the primary digital signature algorithm.³²

G. PASSWORD/LOGIN SYSTEMS

The purpose of password/login systems is identification and authentication. An individual tells the system who he is (identification) and the system proves that the individual is who he says (authentication). Passwords are currently the most common method for authentication of a user.³³ Other authentication devices such as tokens or ID cards are supplements to a conventional login/password sequence.

For every system, there is a need to maintain authentication data in a stored area, such as a password file. Protection of this file is extremely critical in maintaining the security of the system against intrusion. Most computer systems will encrypt the password file. If access security into the password file is breached, encryption still renders the file useless unless the decryption key or code is known to the intruder.

Most systems perform a one way encryption of passwords; this means that the password is never decrypted. When an

³²Kent, S., "Privacy Enhancement for Internet Electronic Mail: Part II Certificate-based Key Management", Network Operating Group, RFC 1422, February 1993.

³³Russell, D., Gangemi, G.T., "Computer Security Basics", O'Reilly and Associates, Inc., July 1992, pgs 58-66.

individual logs in with a password, it is immediately encrypted and compared to the stored encrypted version of the word. If it matches, authentication is completed.

In order to deter encrypted passwords from being deciphered, many systems will incorporate the use of a "shadow" password file which contains the highest security on the system and is accessible only by the system administrator. Any password which is stored for any length of time in an unencrypted form is a security risk to the whole system.

Traditional password models use an identification number and a password and typically follow any one of the following procedures:³⁴

1. Simple password user authentication mechanism: The user inputs, to the system, his identity or a specific identifying number. The user then inputs to the system his secret password. The system compares this information to its own protected identification and password files and attempts to match the identity as registered and the password as valid. This model relies on the security of the users in keeping their own individual passwords and identities private.

2. Variable Password user authentication based on a password list: The user inputs to the system his identity, followed by the user inputting to the system the proper password sequenced from a privately stored list. The system matches the identity and the sequenced password from its own corroborating list as valid. With this method, a password may only be useful once, but the user must memorize and keep track of a list of passwords and the sequenced used. Also, if the sequence of passwords were not chosen at random, the entire user list may be able to be developed by disclosure of a few of the passwords.

³⁴Muftic, Sead, "Security Mechanisms for Computer Networks", Ellis Horwood Limited, 1989, pgs 79-83.

3. Variable password user authentication based on a one way function: The user inputs his identity and private password into the system. The system, using the password as an input, computes an output from a secure function referenced by the individuals identity. If the output from the function is matched and accepted by the system, then the user is considered validated.

In general, every user should have an authentication number and password of his own to facilitate the use of different system access rights, and for system administrators to monitor illegal access by using an audit trail to identify under which identity the violation occurred.

A password system is an appropriate measure for authentication in many situations, but it is fairly weak and subject to attack from many areas.³⁵ Encrypted passwords may be stolen and attacked methodically. Most user selected passwords are social or personality driven and are, therefore, possible to derive from individual research of the user and by brute force methods (dictionary search). In short, passwords provide only for simple authentication.

H. FIREWALL SYSTEMS

For the purposes of this paper, a firewall is defined as a gateway computer designed to limit access from one computer network system to another. For example, controlling access

³⁵Malamud, C., "STACKS: Interoperability in Today's Computer Networks", Prentice Hall Inc., 1992, pgs 158-159.

from an outside public source, such as the Internet, to an inside private source, such as a local area network.

The rationale for using a firewall system is almost always to protect a private network against intrusion.³⁶ Intrusion includes access to company resources by unauthorized users, export of proprietary, classified or sensitive information, or the import of a computer virus.

In terms of computer networking, a gateway is a computer that connects two heterogeneous networks by performing some type of protocol translation. In general, a firewall is composed of one or more of the following components:

1. A Screening Router which offers packet filtering and can block traffic between the network and a specific host. Some "firewalls" used in many private networks are nothing more than a screening router. This results in direct communication between users on a private system and a public, wide area system, such as the Internet. Unauthorized intrusion is difficult to detect if the number of users are high and there is no formally administered audit trail. Also, if an encrypted password authentication system is used and the encryption method or password is compromised, then the entire private system is laid fully open to attack. Screening routers are not generally considered to be the best security solution

³⁶Ranum, Markus J., "Thinking About Firewalls", Trusted Information Systems, Inc., Glenwood Maryland.

but are used extensively since they offer free access to a public network from any point on the private system.

2. A Bastion Host is identified by the system administrator as the single critical strong point in accessing the private networks security system. It is monitored extensively and should undergo regular audits.

3. A Dual Homed Gateway is a bastion host which is placed on both the private system and a link in the public network. It communicates with the public network and the private system, but direct traffic or access from the public to the private system is blocked. The main security risk is the gateway itself, since it is the primary attachment point to the public network. A system administrator could detect and track the progress of an intruder via a dual homed gateway. Gateways have an advantage over screening routers in areas of system independent software maintenance and upgrades, and in the utilization of audit trails for security violations.

4. A Screened Host Gateway is the most common firewall configuration and is composed of the combination of a screened router and a bastion host. The screened router connects to the public network and allows only for selected services to the bastion host of the private network.

5. A Screened Subnet is an isolated network situated between the public system and the private network. It is insulated by a screening router and is configured such that both the public and private networks have access to the

subnet, but direct traffic from the public system to the private network is blocked. Some configurations utilize a bastion host and support terminal sessions or applications level gateways.

6. An Application Level or Proxy Gateway are service specific forwarders which operate in a user mode above the protocol level. It is a potential security concern for a firewall system as it represents a "hole in the wall" of the system for user file transfer, such as "SENDMAIL" or FTP services. Application level gateways should be run and monitored through a bastion host.

7. Hybrid Gateways are a combination of firewall systems and make use of several protocols and configurations to limit unauthorized access.

Two general rules in designing and using a firewall or gateway system are as follows:³⁷

Don't allow non-designated users access to the actual gateway machines, provide the gateway services which permit controlled access and ensures all services are generating login/password/timestamp information on which to base an audit trail.

³⁷Ranum, Marcus J., "A Network Firewall", Digital Equipment Corporation, Washington Open Systems Resource Center, Greenbelt, MD., June 12, 1992.

I. ISSUES ON AUTHENTICITY OF DATA AND USERS

In network communications, authentication provides a way to verify the origin of data and identity of the user. It is a measure used to verify the eligibility and access constraints on an individual user to data or information levels. It protects against fraudulent use of a system or the fraudulent transmission of information and ensures the basic security principle that the information received was in exactly the form it was sent.³⁸

Passwords, as discussed previously, are one area which authentication could be based upon.³⁹ Other areas include the following:

- Pre-arranged information which the user possesses.
- Hardware components such as a physical key, magnetic strip or smart card.
- A personal feature of the user, such as a finger print or retinal scan.

Peer-entity authentication is utilized when the system must authenticate the user and the user has no requirement to authenticate the system. It generally uses a password model in which the user provides an identity and authenticates through a corresponding password. A handshake model may also

³⁸Russell, D., Gangemi, G.T., "Computer Security Basics", O'Reilly and Associates, Inc., July 1992, pg 405.

³⁹Muftic, Sead, "Security Mechanisms For Computer Networks", Ellis Horwood Limited, 1989, pgs 78-79.

be used, which involves a procedure known only to the user and the computer system. When the user logs in, a random number is sent to the user who computes an answer based on a predetermined function or procedure. This methods allows for no secret information to be transferred and does not require the use of encryption for the user response. The limitation of this system is the complexity of the function which the user must perform to verify himself. It must be difficult enough not to be easily detected but simple enough not to be cumbersome.

Peer to Peer or Mutual Authentication is used when two communicants want to authenticate each other. The purpose is to provide a high degree of assurance that a connection has been established with the intended party and not a masquerade. The Handshaking model is preferred for mutual authentication since no sensitive information is shared until the authentication is completed.⁴⁰

Peer to Peer authentication requires two stages; a key establishment stage and an authentication stage. The key establishment stage utilizes interactions between a key distribution center and the users to establish the public or private keys for communication. The authentication stage (handshake) is then conducted by the users and message transfer begins.

⁴⁰Ibid, pg 83.

Message authentication is often ensured through the use of digital signatures. The digital signature is a series of bits attached to the message based upon the message content and the sender's identity or key. Digital signature models are based on both symmetric and asymmetric encryption standards.

III. DISTRIBUTED SYSTEMS AND SERVICES

In this chapter a discussion of the third research question is made: What are some of the currently available distributed file systems and services? A short description is given of each system with respect to the following areas: system design goals, a brief theory of operation, and authentication, security and file sharing and replication issues. This chapter concludes with a discussion of common characteristics of the reviewed distributed file systems and what characteristics a single DoD wide system should incorporate.

Currently available systems under consideration are:

- A. The Sun Network File System
- B. The Andrew File System
- C. The Coda File System
- D. The Apollo Domain File System
- E. AT&T Remote File Sharing
- F. SPRITE Network File System
- G. IBM AIX Distributed Services
- H. The LOCUS Distributed File System
- I. Project Athena and Kerberos
- J. The FICUS Distributed File System
- K. Common Characteristics of Reviewed Distributed File Systems

A. SUN NETWORK FILE SYSTEM (NFS)

1. System Design Goals

Introduced in 1985 by Sun Microsystems, the Network File System (NFS) is one of the most widely used file transfer system today.⁴¹

Primary considerations in the design of NFS are in areas of portability and heterogeneity. Although initially designed for the UNIX system, versions have been utilized on PC-DOS machines. This type of portability is possible because, during the design of NFS, Sun Microsystems kept the NFS protocol and the implementation schemes separate. The protocol defines the interface which allows the server to export local files but does not dictate how the server should implement the interface or its use by the client.

The implementation of NFS defines the extent of the file caching, replication naming and consistency. The interface protocol was designed to be "stateless" where the request of the server made by the client contains all the necessary information which the server requires about the client. This information, therefore, does not need to be maintained by the server.

⁴¹Cohen, D., "AFS: NFS on Steroids", LAN Technology, March 1993.

2. Theory of Operation

Initially, NFS treated all workstations as peers where the roles of client and server could be performed by any machine. Despite this design it is more common for system administrators to configure local installations with a limited number of workstations dedicated solely as servers and the remaining workstations as clients.

Each file name with NFS has a "private root" which acts to bound the file to nodes within an administrative file system. Each workstation, in manipulating the file, configures its own namespace on the file in conjunction with its private root.

An NFS client will cache remote files, directories, and path-name translations into main memory and not onto a local disk. Along with the cache of the file, a timestamp is also cached which indicates when the file was last modified on the server. This timestamp is used for a "latest copy" validation by comparing the cached timestamp with the file timestamp. If the cached timestamp is more recent, the file timestamp is considered invalid and the client will re-cache the current file from the server. This validation check is done at each file opening. Directories are cached in a manner similar to files with the exception that any modification is performed directly on the server.

Data transfers for NFS are normally done in block sizes of eight kilobytes. Files are fetched in entirety if

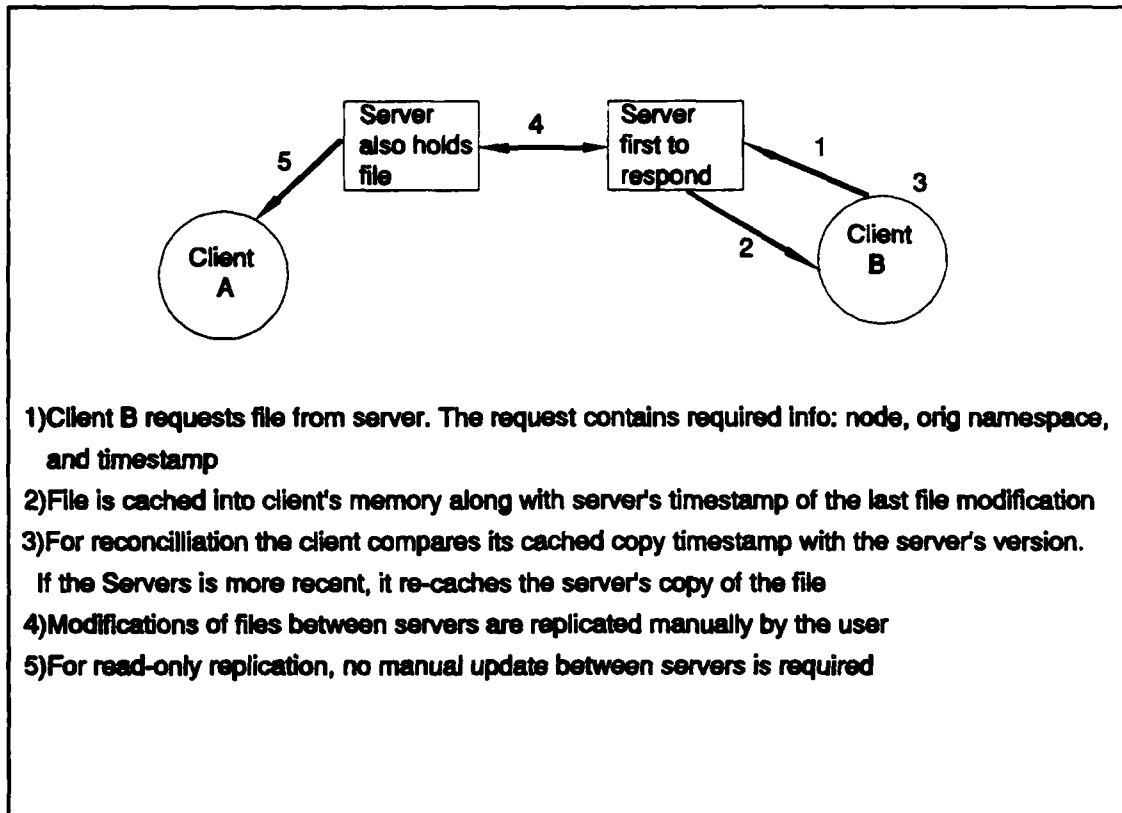


Figure 2. SUN Network File System

they are smaller than this threshold.

Current versions of NFS support replication via an "automounter" mechanism where remote points for file accessibility are specified using a set of servers. When a client issues a request for a file, the first server to respond is chosen as the remote automounter point. Any additional requests which use this remote point are directed to its server. This replication mechanism is intended primarily for read-only vice write. Because of this (read-only), all propagations of file modifications of file replicas held by other servers must be performed manually on those servers by the user.

3. Authentication, Security and File Sharing Issues⁴²

NFS uses underlying UNIX mechanisms for file protection and system access. It can also be configured to provide for a high level of security based on the DES encryption algorithm for authentication and client/server validation at each request. Public/private key encryption techniques are also supported to provide for mutual authentication of the DES key.

One complication of the stateless NFS protocol is reduced functionality with respect to file locking during multiple access on a distributed system within the UNIX domain. File locking allows for only one user modification on the file at a time, where the file is "locked out" from others who wish to modify the file at the same moment. A lock on a file constitutes state information on the server (The server has to block access to the file). In a UNIX system this is accounted for by the Sun operating system which provides for a separate "lock server" to perform this operation.

Another issue concerning file replication and location transparency involves workstation namespace identifiers. Since each workstation is free to configure its own namespace, there is no guarantee that each workstation will utilize a common view of a shared file or that naming conflicts won't occur.

⁴²Satyanarayanan, M., "A Survey Of Distributed File Systems", School of Computer Science, Carnegie Mellon University, Annual Review for Computer Science, 1990.

This situation is generally overcome by system administrators who collaborate with other similar groups to configure their workstation's namespace in a similar manner. As a result, location transparency is enforced by personnel convention rather than NFS architecture.

Remote file access and replication is supported via two types of services. The first is an automounter (as previously stated) for replication of read-only files. The second is through the use of "Yellow Pages" which provides for a listing of specific applications, user-names, and passwords, host names and network addresses, and network services to Internet port numbers. The Yellow Pages provide for read-only replication by utilizing one master file and several slave files. Look-ups (read-only) can be performed on any replicated slave but modifications can only be performed on the master file. Any modification to the master file must be manually propagated to the slave files.

B. THE ANDREW FILE SYSTEM (AFS)

1. System Design Goals

The Andrew File System was developed by Transarc Corporation of Pittsburgh, Pa. It is a Client-Server computer network system which provides for authentication and authorization by combining the functionality of the NFS with

a Kerberos style ticket authentication scheme which has been augmented with access control lists (ACL's).⁴³

AFS is an attempt to combine a user interface of personal computing with the data sharing capabilities of time sharing systems.

2. Theory of Operation

The AFS consists of a distributed collection of servers known collectively as "Vice". These servers service a large number of workstations which run a collection of software known as "Virtue". All files are stored permanently in the vice servers. Files are maintained temporarily in the virtue clients by extensive caching. A workstation installation will use a global name for its users identifiers.

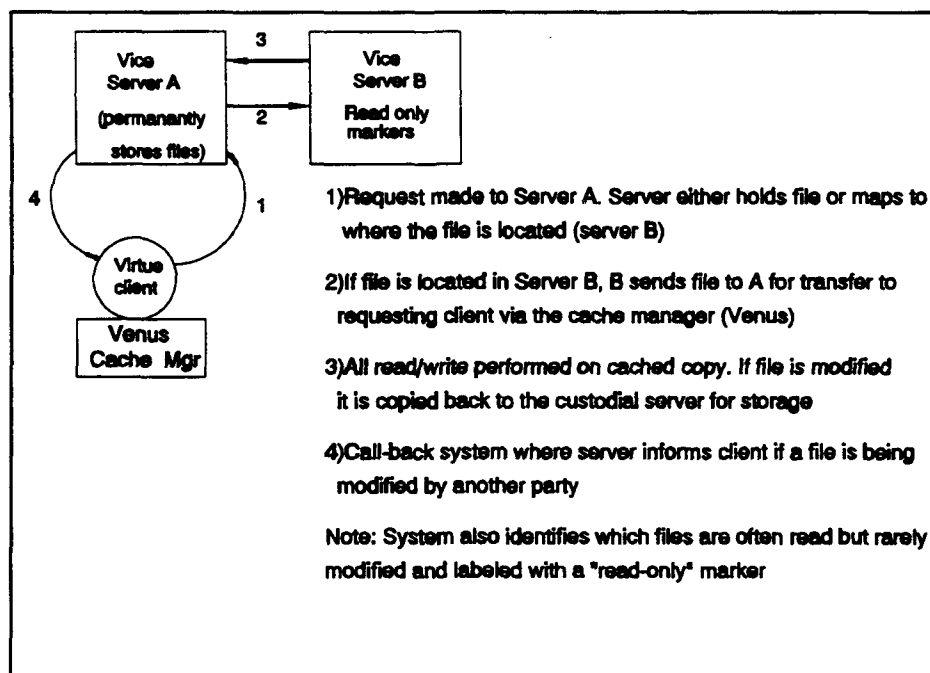


Figure 3. Andrew File System (AFS)

⁴³Arnold, N., "UNIX Security: A Practical Tutorial", McGraw-Hill, Inc., 1993, pg 191.

Scalability is the dominant design consideration in AFS. It has an anticipated final size of approximately 10,000 nodes.⁴⁴ File name spacing is partitioned into a shared local namespace for that file at the workstation center. The shared namespace is location transparent to the user and is identical on all workstations. The workstation contains only temporary files or those specific files necessary for workstation initialization. Data imaging for the users is consistent at each workstation since their files utilize a shared namespace.

The namespaces themselves are structured in a hierarchical order. They are partitioned into subtrees which are assigned to a single server called a subtree custodian. Each server also contains a copy of a fully replicated location database which maps files to the subtree custodians. This database maintains size manageability since the custodianship pertains mainly to the subtrees vice individual files.

Files in a shared namespace are cached on local disks of the workstation which are controlled by a cache manager called Venus. When a user opens a file, Venus will first check the workstations cache for the presence of a valid copy. If one is there, the request is treated as a open file. If it is

⁴⁴Satyanarayanan, M., "A Survey of Distributed File Systems", School of Computer Science, Carnegie Mellon University, 1990.

not, Venus will fetch a copy from the custodian of the file. All read and write operations are performed only on the workstation's cached copy. If the cached copy is modified, it is then copied back to the custodian of that file when the workstation file is closed.

Cache consistency between users is maintained via a call back system where the server informs the client if the file is being modified by another party. File availability is increased by identifying which files are most often read but rarely modified as labeled with a "read-only replication" marker.

Concurrency control of a file is the ability to control operations on that file by more than one workstation simultaneously. It is provided for with timing lock and unlock operations by the system administrator or file custodian. If a lock on a given file is not released by the client within a specified time (for example, 30 minutes) then it is timed-out by the server and becomes unlocked for other clients.

3. Authentication, Security and File Sharing Issues

Unlike Kerberos, the AFS performs mutual authentication between the workstation and the Vice servers at the beginning of the communication process. After the first dialogue, subsequent communications can be encrypted or merely authenticated. File replication on a permanent basis between work sites does not occur since local copies of the files are

cached only. Only workstations which are connected via the AFS installation can share files. Users or sites outside the AFS system cannot share server files.⁴⁵

System security is predicated on the integrity of the Vice servers, which are kept physically secure and only accessible by trusted system operators. Both the workstations and the network itself are assumed untrusted by the servers; therefore, all secure transmissions rely solely on end to end encryption.

AFS uses access control list mechanisms for user protection. All system privileges endowed upon the user are specified either directly to him or the groups to which he belongs. AFS also allows for "negative access rights" which indicates a denial of selected services to the user.⁴⁶

AFS authentication mechanisms are compatible with and can be supplemented by Kerberos authentication mechanisms of Project Athena, developed at the Massachusetts Institute of Technology. They resemble each other in terms of architecture and both use similar authentication schemes.⁴⁷

⁴⁵Reiher, T. Page, G. Popeck, "Truffles - A Secure Service for Widespread File Sharing", Dept. of Computer Science, University of California, Los Angeles, pg 110.

⁴⁶An Orange Book "B" level classification requirement.

⁴⁷Satyanarayanan, M., "A Survey of Distributed File Systems", Annual Review for Computer Science, 1990, pgs 73-104.

The data structure used by AFS is built around a collection of files and directories in the Vice namespace of a similar custodian called a "volume". There is usually one volume per user and volume sizes are kept small enough to allow several partitioned volumes on each server. The volumes also form the basis for the backup and restoration processes. Backups are created by cloning the files and then sending the clones to a staging machine for transfer onto a storage media.

Authentication and translation of users from different administrative domains are coordinated by the use of cells. Cooperating cells adhere to a standard set of protocols and naming conventions which provide the user with the image of a single file namespace.

C. THE CODA FILE SYSTEM

1. System Design Goals

The Coda File System was designed for a UNIX workstation based on a large scale distributed computing environment. It provides for server replication by storing file copies on multiple servers, and disconnected operation which is an execution mode where the caching site temporarily assumes the role of a replication site. Disconnected operation was designed to enhance the use of portable workstations in a distributed processing environment.⁴⁸

⁴⁸Ibid.

Other Coda design goals included maximizing availability and performance, and maintaining a high degree of consistency within the file replication structure. The system is location transparent and is based on the UNIX file system model.

Coda is an extension of the capabilities of the Andrew File System (AFS) and is an attempt to maintain the functionality, performance, and administrative ease of AFS, but attempts to limit its mechanical vulnerabilities to server or network component failures in a large area file sharing environment.⁴⁹

The main design goal of Coda is constant data availability which allows access to data regardless of extraneous failures elsewhere in the system.⁵⁰ A secondary goal is to allow for the integration of AFS on portable computers. The system was designed to emulate UNIX file semantics and be highly scalable while automatically accommodating data access despite a wide range of system failures.

2. Theory of Operation

Coda retains many of the same features as AFS: a model composed of a few trusted and protected servers with several

⁴⁹Satyanarayanan, M., et al, "Coda: A highly Available File System for a Distributed Workstation Environment", IEEE Transactions on Computers, Vol 39, No 4, April 1990.

⁵⁰ibid.

untrusted clients. It is also similar to AFS in terms of client caching of entire files onto local workstation storage media (disks), and the use of coordinated system callbacks to administratively maintain cached file coherence.

For replication Coda operates differently. It uses the server for replication where files are copied and stored on multiple servers. This allows for a higher file availability rate than AFS provides. The client relies on the server for replication as long as the client maintains contact with at least one server. If no server can be contacted, then a "disconnected" operation occurs where the client relies solely on its cached version of the file. Coda achieves higher availability than AFS by the replication of files across several servers (vice a single custodian for AFS), and by allowing for clients to operate entirely out of their cache if a server is not connected.

The unit of replication in Coda is a volume: a set of files and directories located on one server and forming a partial subtree of a shared namespace. Each file and directory has a low level file "identifier" which identifies itself to a parent volume. All replicas of a file contain the same file identifier.

The set of servers with replicas of a volume constitute a Volume Storage Group (VSG). A volume replication database is also present in every server and administratively maintains the amount of volume replication and its associated

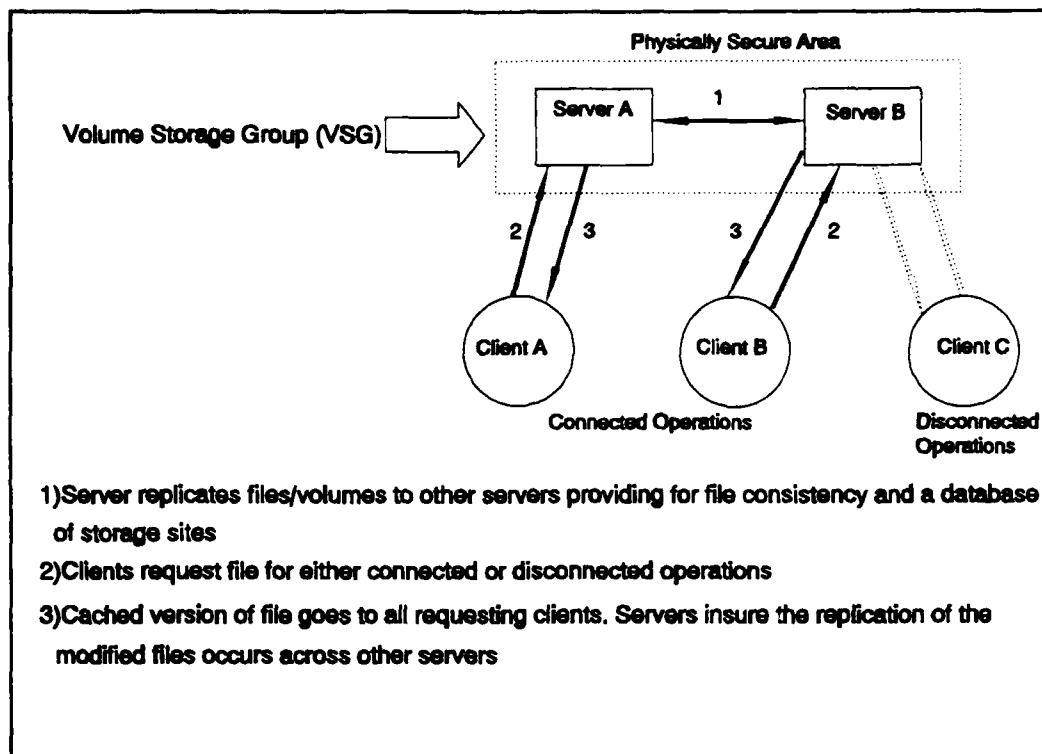


Figure 4. Overall View of Coda

storage site. A client manager (called Venus in AFS) keeps a list of the volume storage groups for every volume from which it has cached data. This list is called an Accessible Volume Storage Group (AVSG) and each client maintains a copy of this separate group listing.

The replication strategy uses a read-one, write-all approach. When a request is made from a client to a server, the client will also contact other servers to ensure the copy being received is the latest version utilizing a timestamp on the file. If not, a listed member of the client's AVSG with the latest copy (a more recent timestamp) of the file becomes the preferred retrieval site where the file is then retransmitted.

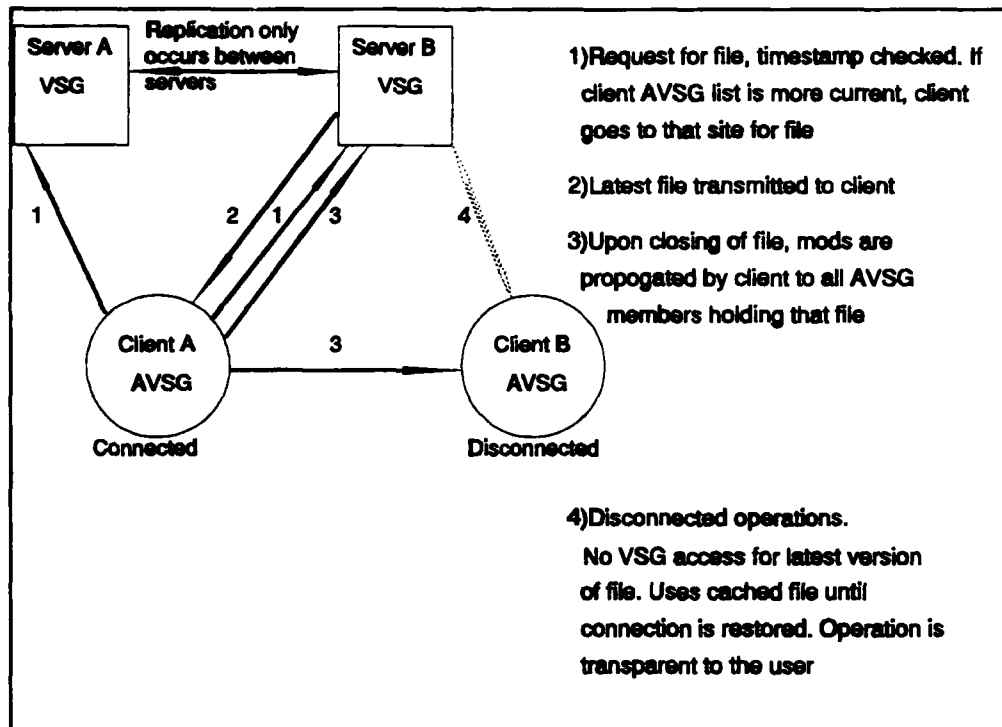


Figure 5. Coda in Connected and Disconnected Modes

After the file has been modified, it is closed and transferred to all AVSG members. With this method of file propagation, the burden is on the client's processor rather than the server. This reduces the processing burden of the server and, consequently, eliminates a typical bottleneck of many other types of distributed file systems.⁵¹

Disconnected operation is initiated by the client when no member of a Volume Server Group is available. It is transparent to the user unless an error in his cache occurs, in which case, the user is unable to continue until normal connected operation is resumed. Unless a conflict is detected,

⁵¹ibid.

then the automatic return to normal operation is also transparent. When the disconnected operation ends, the client manager reintegrates the client's cached file by executing a sequence of update operations to make client AVSG replicas identical to the previously disconnected cached copy. Conflict resolution is generally automatic. If Coda cannot automatically solve an update conflict between replicas, a system administrator must manually resolve the discrepancy.

3. Authentication, Security and File Sharing Issues

Coda is an attempt to incorporate the virtues of AFS in a large scale file distribution system with higher availability. Coda is also similar to the LOCUS Distributed Operating System in that both use UNIX directory semantics to automate conflict resolution between replicas. However, unlike LOCUS, file security was a fundamental goal designed into Coda, it supports a token-based authentication scheme and end-to-end data encryption methods.⁵²

Coda is based on a client/server model and incorporates the use of portable computers through disconnected operation and whole file caching and replication. Clients must update all server replicas and users communicate to Coda through an approximation of UNIX semantics.

Coda is specifically designed for a workstation environment. It uses caching to reduce network and server

⁵²ibid.

load, supports user mobility and system scalability. Replication is used only among servers. Clients keep track only of the servers and not of other clients. Modifications to the server replicas are performed by the clients which reduces server workload.

D. APOLLO DOMAIN FILE SYSTEM

1. System Design Goals

The Apollo Domain File System (ADFS) was built by Apollo Computers Inc. in 1980 for a distributed workstation environment. The purpose of the system was to provide an efficient computing base for a moderately sized group of collaborating individuals. Unlike AFS, scale was not the dominate design consideration. The underlying network technology is a proprietary token ring where some nodes act as servers and the rest as clients. This methodology is only for user convention as ADFS treats all nodes as peers.

The goals of the system are location transparency, data consistency, a system enforced uniform naming scheme, and a uniformed method of access control. Other administrative goals of the system are full functionality, good performance and administrative ease.⁵³

⁵³Satyanarayanan, M., "A Survey of Distributed File Systems", Annual Review for Computer Science, 1990.

2. Theory of Operation

ADFS utilizes an Object Storage System (OSS) to support the distribution of files. On top of the OSS is a Single Level Store (SLS) which provides for a mapped virtual memory interface for the objects.

Each object within the system is named by a User Identification (UID) marker. UID's are a 64 bit string composed of a unique creating workstation number and a timestamp.

Every object has a home node which the OSS maps through a "Hint" server. A "Naming" server maps string names to UID's and provides a hierarchical, UNIX-like, location transparent namespace for all files and directories within the system. Directories within ADFS are objects which map name components to UID's. A network wide root directory of the file namespace is implemented as a replicated distributed database with the server located as the site of each replica.

A timestamp indicating the last modification is associated with each object and every cached object contains the timestamp. Object consistency is maintained by comparing cached object timestamps with the original object timestamp in the home node. Any object found invalid is automatically discarded.

Object concurrency control is integrated with cache management via a node driven "lock manager" which synchronizes accesses to any object within the home node's domain. The lock

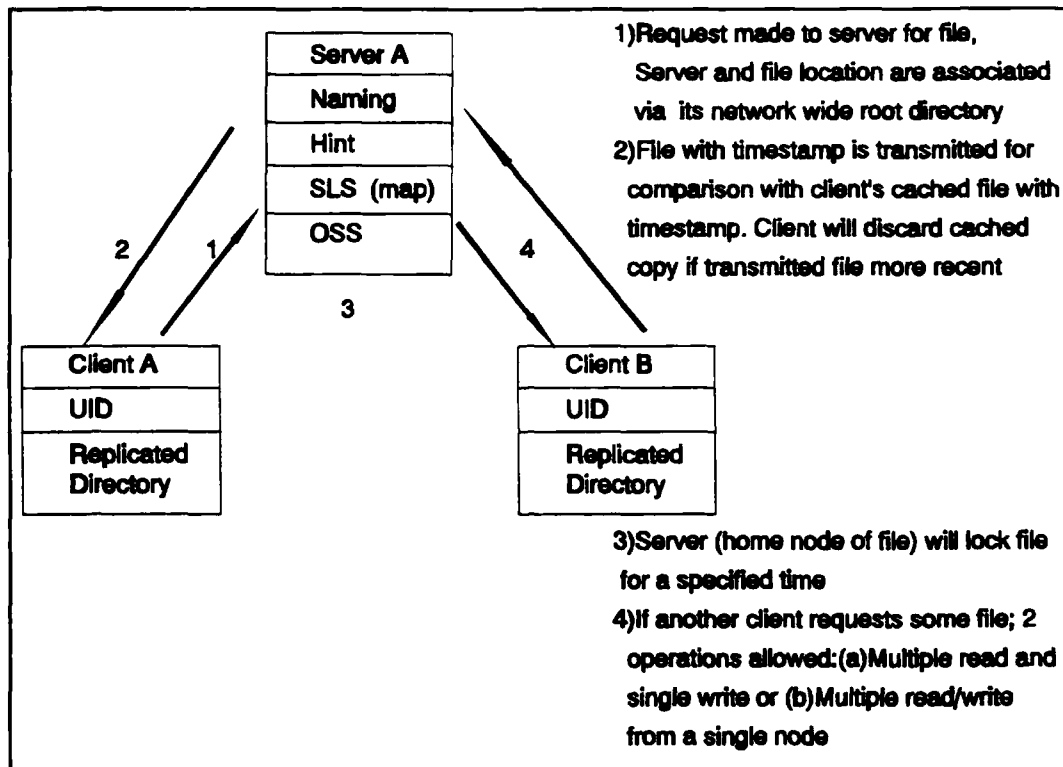


Figure 6. Apollo Domain File System

manager allows for two type of operations: multiple distributed readers and a single write access to an object, or access for multiple readers and writers located at a single node.

3. Authentication, Security and File Sharing Issues

ADFS does not support read-only or read-write data replication. There can be only one home node for any object at any time.

Security of ADFS relies on the physical security of the workstations and the security of the operating system kernels within them. Since the network is assumed to be secure, data transmissions between nodes are not encoded.

Provisions within the kernel are made to prevent user programs from masquerading as trusted system software.

ADFS supports encrypted password login methods for validation of the user to the system. User registries are a replicated database with one master site and multiple read-only sites for availability.

ADFS supports a decentralized administration regime where different groups use a single registry for system management. This allows each group to engage its own system administrator who has sole access for manipulation of any entries pertaining to that group.⁵⁴

E. AT&T REMOTE FILE SHARING

1. System Design Goals

Remote File Sharing (RFS) is a distributed file system developed by AT&T for its System V version of UNIX and uses a precise emulation of local UNIX semantics at local sites. Any operation on a remote file appears identical to any operation on a local file. This design philosophy extends to other mechanisms within RFS; concurrency control, write sharing semantics, and access or control of remote devices. Other design goals include system portability and accuracy.

⁵⁴ibid, pgs 82-85.

2. Theory of Operation

RFS is based on client server relationships and uses UNIX System V mechanisms for portability across varied transport protocols. The server "advertises" which files or subtrees it wishes to export with a network wide symbolic name. The client imports the symbolic name and uses a naming server to translate the symbolic name to a server address of the file.

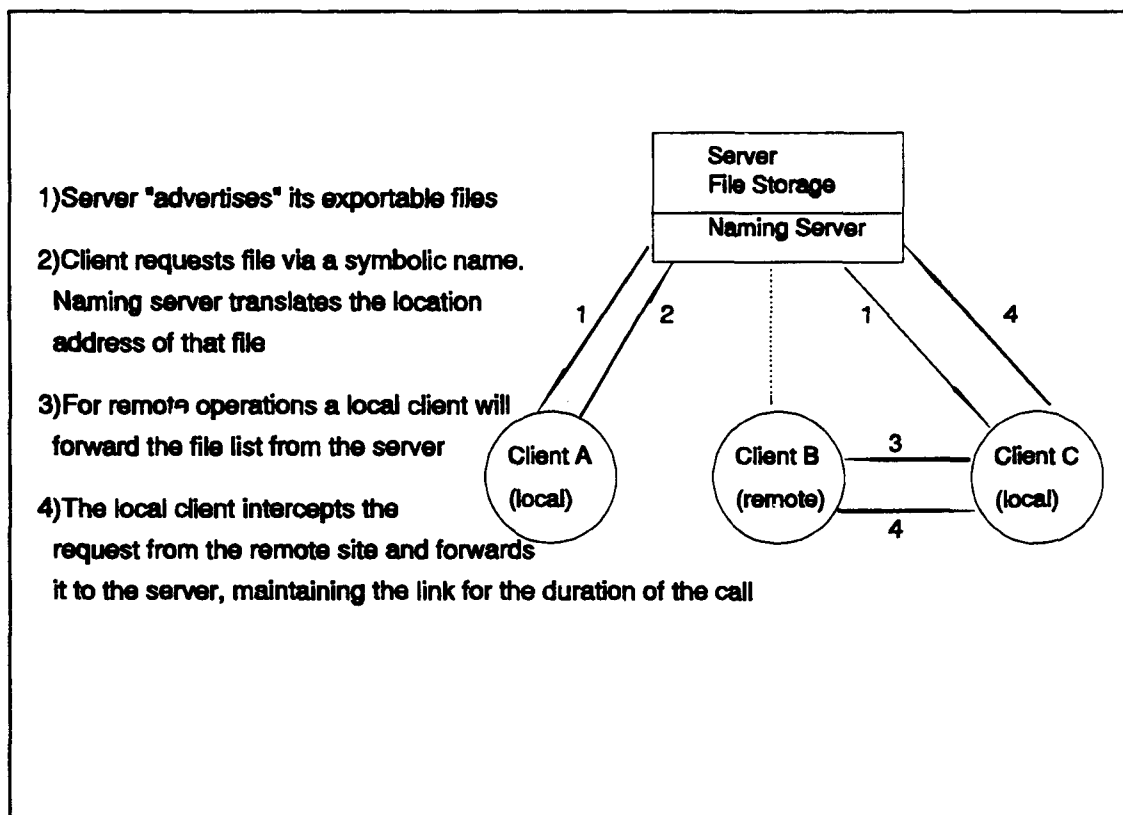


Figure 7. AT&T Remote File Sharing

Accuracy is maintained by using the server for all remote file system calls. A client will intercept a call, forward it to the server and then recreate the request to the server for the duration of the call.

Information caching is maintained in the clients main memory and is used only for read-only of simple files and not directories or devices. Consistency of the file is checked when it is opened. When the situation is a single writer and multiple readers, then client caching is disabled until the writer closes the file. If there are multiple writers, then all caching in the system is disabled.

3. Authentication, Security and File Sharing Issues

Protection of RFS files and directories are specified exactly as in UNIX, client and server implicitly trust each other. RFS also provides for mechanisms to map user and group identities for shared access across administrative domains with allowances for restriction privileges of remote users at local sites.⁵⁵

RFS is similar to NFS with the major difference being that NFS transport protocol is stateless. RFS maintains a server state for file operations. RFS does not readily support file replication or protocols for establishing a replicating, sharing relationship.⁵⁶

⁵⁵ibid, pgs 90-91.

⁵⁶Reiher, P. et al, "Truffles - A Secure Service for Widespread File Sharing", pg 110.

F. SPRITE NETWORK FILE SYSTEM

1. System Design Goals

The Sprite Network Operating System (SNOS) was developed for workstation operations by the University of California at Berkeley. There were six major design goals of SNOS: efficiently use workstation main memory capacity, support multiprocessor workstations, ensure efficient network communications, provide for diskless operations, emulate UNIX file system semantics, and provides for distributed file system facilities such as process migration.⁵⁷

2. Theory of Operation

SNOS was designed to be a diskless system with no real distinction between server and client; however, the implementation of SNOS incorporated a few dedicated servers with disk storage systems. The interface system from the servers uses a location transparent UNIX file mechanism to the clients.

Servers respond to client file requests by using "remote links" or pointers which are embedded in the file system at each server. Clients maintain a local prefix table which maps pathname prefixes to the servers.

The server is notified whenever a client opens or closes a read-only file or initiates a read-write file

⁵⁷Satyanarayanan, M., "A Survey of Distributed File Systems", Annual Review for Computer Science", 1990, pgs 91-92.

operation. The client will cache the file and validate it based of the time the file was opened. If one or more clients open a file for reading and writing, caching is disabled and is re-enabled only after all the clients concurrently using the file have closed it. This method allows for maintaining the validity of the files and ensuring internal consistency of read-write operations.

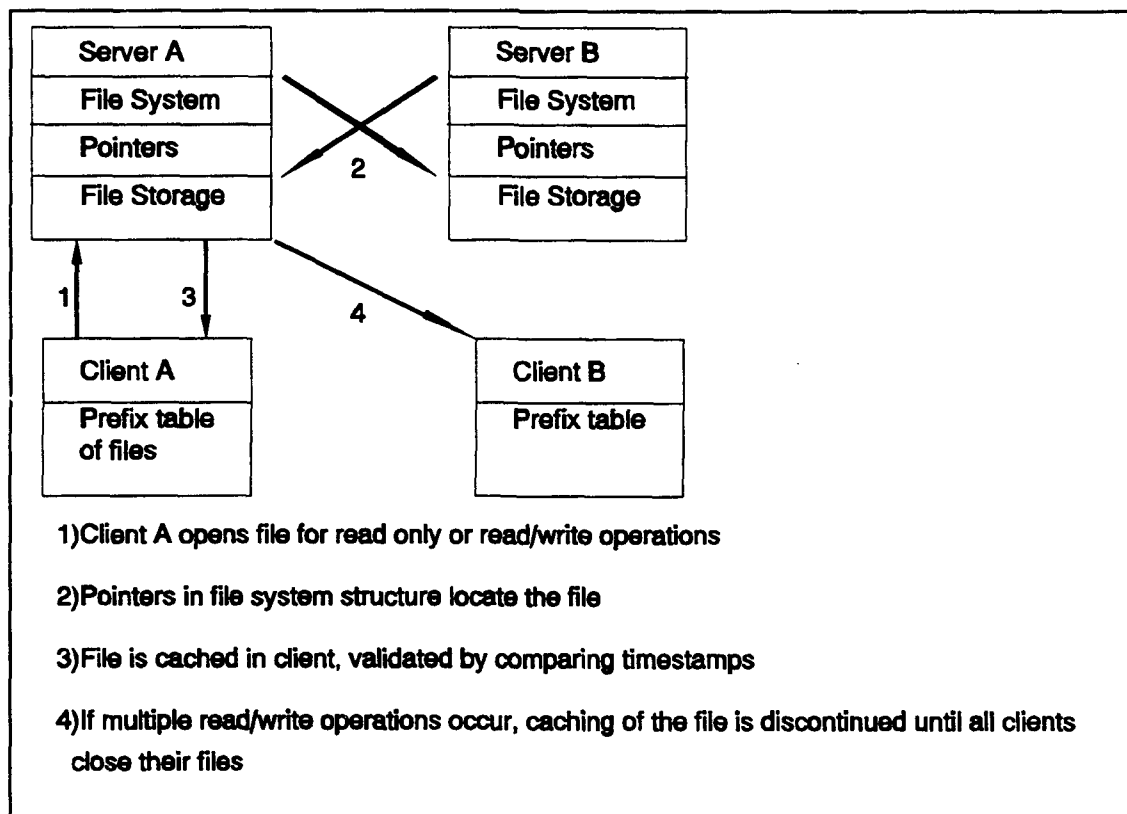


Figure 8. Sprite Network File System

3. Authentication, Security and File Sharing Issues

SNOS was designed for use by collaborating users or groups who are unable to subvert or reconfigure the workstation operating system kernels, or who just implicitly

trust each other. A basic assumption of the SNOS system is that SNOS kernels trust each other and network communications are not authenticated or encrypted.

SNOS provides for location transparent remote access to devices as well as files. High workload performance is enhanced by dynamically partitioning workstation main memory into a virtual memory subsystem and a file cache. SNOS utilizes shared name-spacing which allows for simplified process migration between servers and workstations and common user views at any workstation within the network.

G. IBM AIX DISTRIBUTED SERVICES

1. System Design Goals

IBM developed the AIX Distributed Services (AIX-DS) as a collection of different distributed services for implementation on top of its workstation AIX operating system. AIX is an emulation of the UNIX operating system and the AIX-DS utilizes an emulation of UNIX file sharing semantics. Other important aspects are its efficient support of local database configurations and varied administrative distributed system installation configurations.⁵⁸

2. Theory of Operation

AIX-DS uses the UNIX mount mechanism to access remote files and allows individual files and directories to be

⁵⁸ibid, pgs 89-90.

mounted directly to a server without the requirement of mounting the entire directory subtree. All files are assumed to be accessible and most file system operations behave similarly on local or remote sites.

AIX-DS uses a client-server relationship where the client informs the server at each file opening operation. If the file is accessed in a read-only mode, then file caching is allowed simultaneously to multiple clients. If the request is made for one client to read and write, then the file access is in an asynchronous mode where file caching is enabled only to the writer. A third mode, full synchronous, is enabled when multiple clients require writing to a file. In this situation, client read-only caching is disabled and file consistency is maintained in a manner similar to the Andrew File System. The server keeps track of all client openings of its file and verifies the time it was modified. It informs the clients of the modification, the clients then invalidate the copies of the files in their caches and can receive the modified copy.

3. Authentication, Security and File Sharing Issues

AIX-DS is based on the SNA LU6.2 protocol and can support the more common TCP/IP protocol. DES encryption for node to node communication and for mutual authentication is also supported, as well as the Kerberos authentication mechanisms.

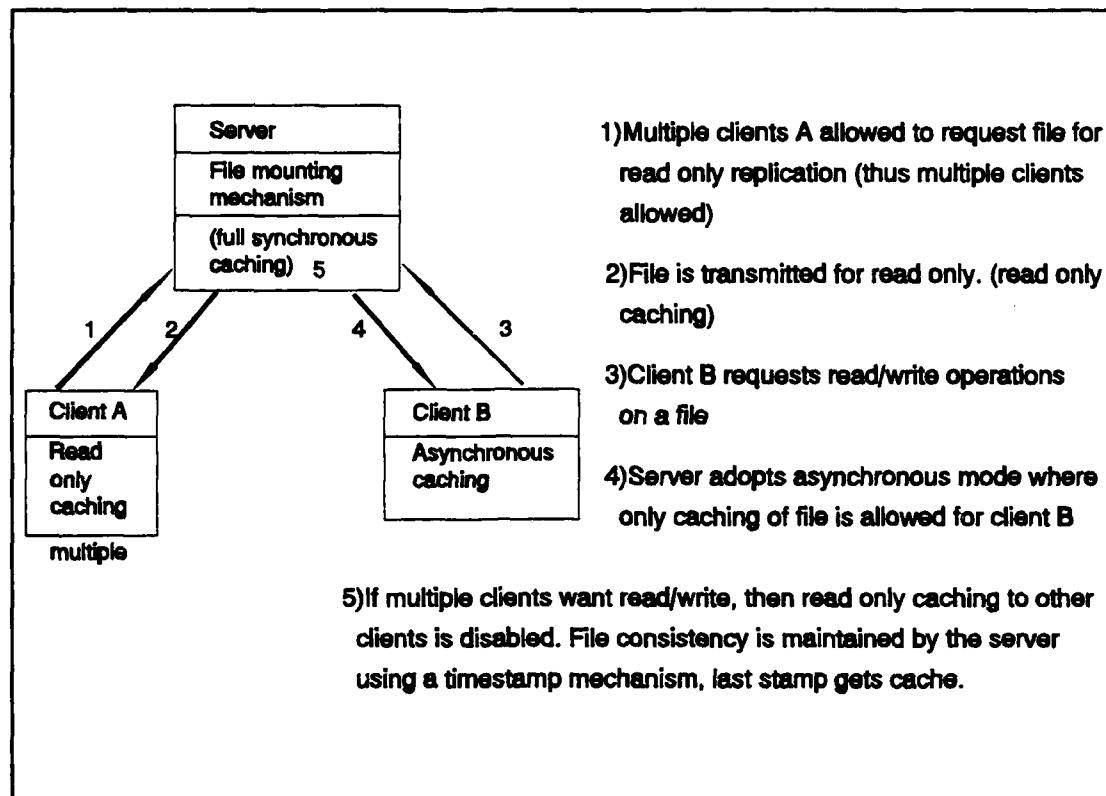


Figure 9. IBM AIX Distributed Services

Within the network, users and groups utilize 32 bit identifiers. AIX-DS translates these into UNIX compatible machine specific 16 bit identifiers for access outside the local network.

H. THE LOCUS DISTRIBUTED OPERATING SYSTEM⁵⁹

1. System Design Goals

LOCUS is a distributed operating system which was designed at UCLA under an ARPA contract.⁶⁰ It was designed to be a network wide file system with the ability to support transparent access to data, automatic replication and distributed process executions in a UNIX compatible environment.

The system was designed to support, to a high degree, network transparency to make the separate network computers appear to the users and compatible applications as a single machine. Use of the system for file creation or manipulation and processes execution are functionally the same whether the users access the system locally or remotely. Many of the functions of LOCUS were designed to be operated transparently across heterogeneous machines.

Important to the design of LOCUS is the research conducted on recovery from failures of parts of the system, consequently, several of the LOCUS functions deal with the process of file or system recovery.⁶¹

⁵⁹Walker, B., et al, "The LOCUS Distributed Operating System", Operating Systems Review, Volume 17, Number 5.

⁶⁰Advanced Research Projects Agency (ARPA) contract DSS-MDA-903-82-C-0189.

⁶¹Walker, B., et al, "The LOCUS Distributed Operating System", Operating System Review, Volume 17, Number 5, pg 49.

2. Theory of Operation

The LOCUS file system is a functional superset of the UNIX tree structured naming system. It presents to users a single tree structured naming hierarchy. The tree structure covers all objects in the file system on all of the machines. It is impossible to discern the particular location of the object from its name alone. This aspect means that LOCUS names are fully transparent and allow for data and programs to be moved or executed to and from different sites.⁶²

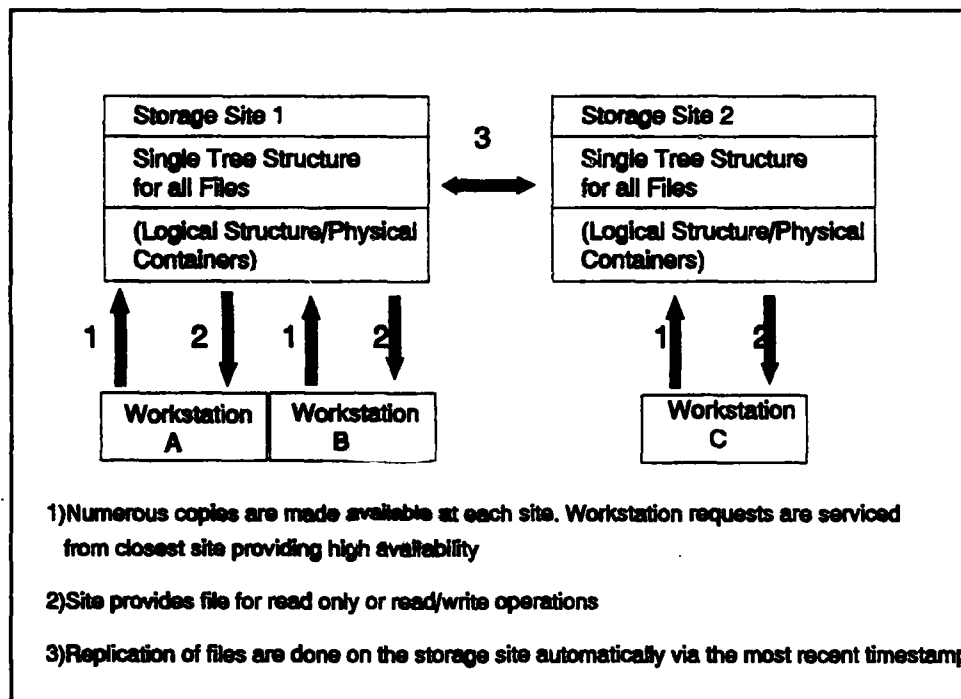


Figure 10. The LOCUS File System

Replication of files is performed on the storage sites and LOCUS automatically ensures all copies are up to date with access made to the most recent version.⁶² High availability

⁶²ibid, pg 51.

is assured by the numerous replicated copies. Performance (or retrieval speed) is enhanced by having the copies close by the requesting machine; in other words, if the users of a file exist on different machines and copies are available near those machines, the access to those files are much faster than if the user had to access all the files remotely.

File replication is through multiple physical containers for a logical file group. A file belonging to a logical group may be stored at any site where there is a physical container corresponding to that group.

LOCUS is designed so that every site is a full functioning node where file operations may involve more than one host. In a file access, there can be any combination of the three logical sites:

1. The Using site which issues the request and receives the file.
2. The Storage site where a copy of the file is stored.
3. The Current Synchronization site which has information about which sites store the requested files, the most current version and enforces global access synchronization in retrieving those files. All requests for a file go through this function, which allow for the implementation of a variety of access policies at a single source.

3. Authentication, Security and File Sharing Issues

The basic approach in LOCUS is to maintain a strict synchronization among all copies of a file so that any user who sees that file sees only the most current version, even if

a concurrent update is occurring on a different machine. Copies maintained in a single partition within the system are the primary source of difficulty in the process of replication. A copy in each partition will operate independently of the other and when they are merged, the system will automatically reconcile the data types which it understands. If it cannot reconcile, the problem is flagged to a higher level, usually a system administrator or to the users who must interactively merge the copies.

Locus is a high performance network transparent distributed file system which has successfully been run in a single administrative domain with all the sites supported in close cooperation. This type of cooperation is possible on a small localized network but may be difficult to achieve with a broader, large area, file sharing network utilized by arbitrary users at other sites where there is no single administrative authority to control the entire system. Also, general security issues, other than those which are provided by a basic UNIX installation, were not considered in the LOCUS system design.⁶³

Implementation of LOCUS is based on a peer-to-peer network model. Replicas are updated at a single replication site, which then notifies other replicated sites of a pending

⁶³Reiher, P., et al, "TRUFFLES - A Secure Service For Widespread File Sharing", Proceedings: Workshop on Networks and Distributed System Security, February 1993.

update. These replicas are then updated asynchronously from the first replication site.

I. PROJECT ATHENA AND KERBEROS

1. System Design Goals

Kerberos is a trusted third party authentication system for open computer systems and networks. It was developed by project Athena in 1988 at the Massachusetts Institute of Technology and funded primarily by DEC and IBM. It is designed to be added into any existing network protocol, although it has been generally used with the UNIX oriented protocols such as NFS.

The system is considered trusted in the sense that each of the users believe Kerberos's identification of each of the other users is accurate. It provides mutual authentication between users on an open local area network.

Design goals included no clear text transmission of passwords or plaintext password storage on servers, minimized exposure of client and server keys where compromises only affect the current session, and limited but re-usable authentication lifetime. It was designed to be transparent to the user and requires minimal modification to existing network applications.⁶⁴

⁶⁴Hughes, Larry J., "A Brief Overview of Kerberos Authentication", Indiana University, University Computing Services Network Applications Group, October 1993.

2. Theory of Operation

Kerberos uses DES and every user maintains a private key. It uses cryptographic keys known as "tickets" to protect the security of messages sent in and out of the computer system. Passwords are never transmitted and reside in a highly secure machine called a key server. It performs authentication on login and network service requests.

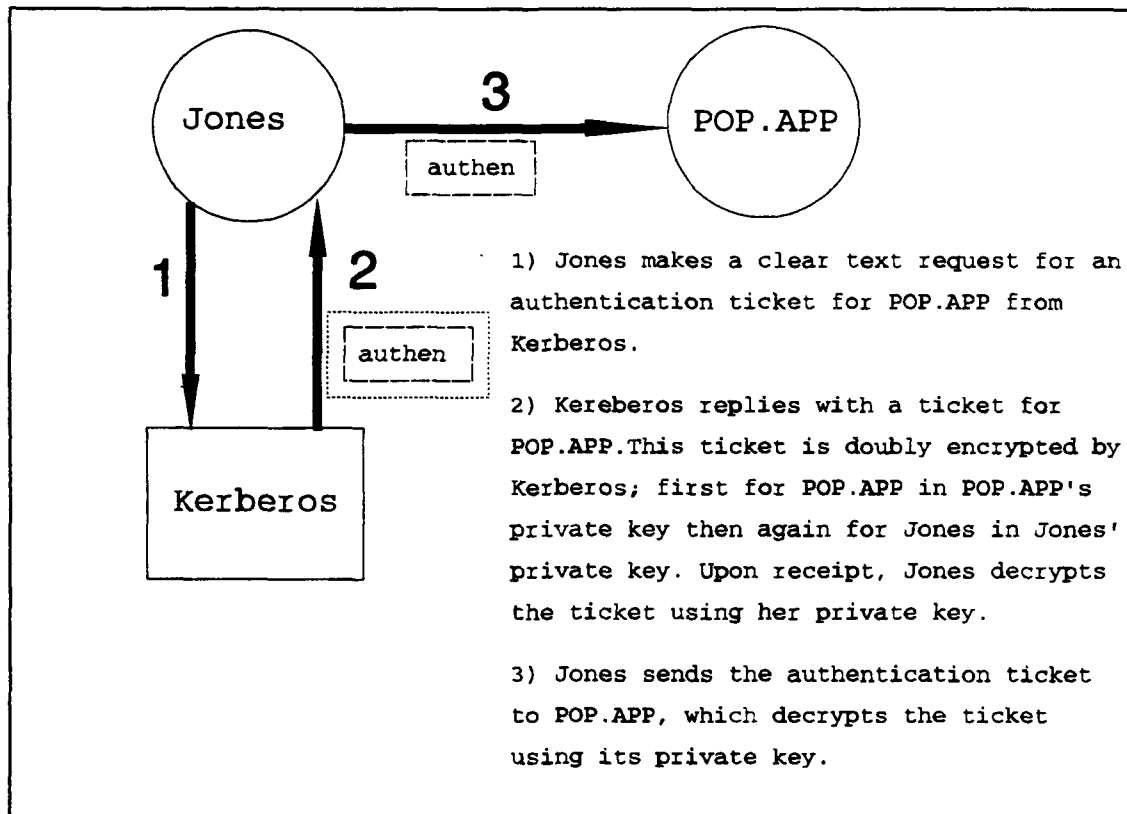


Figure 11. Kerberos Authentication Scheme

In operation, User A makes a clear text request for an authentication ticket for a particular service from Kerberos. Kerberos replies with a ticket for that service which is encrypted first by the service's private key and then by User A's private key. When User A receives the ticket, he decrypts

it using his private key and send the ticket to the requested service, which then performs a final decryption using its private key. If the final decryption is not in the proper form, then User A is not authenticated (an invalid key) and the service is not rendered. In this process, authentication is achieved without ever sending a password over the system. It supports UNIX, VMS, Macintosh and DOS implementations.

3. Authentication, Security and File Sharing Issues

Kerberos' strengths are its external adaptability to heterogeneous systems, passwords are not transmitted, and it provides for a trusted, single centralized authentication authority for a local network. It prevents replays of client server associations and provides for a secure data encryption scheme via DES and can detect a message stream modification by an untrusted source. By providing client and server with mutually agreed upon session keys, it allows for a limited secure data flow.

However, Kerberos does not allow for prevention of dictionary attacks of user passwords or have methods to prevent users from sharing passwords from other users. It does not prevent the denial of service attack and does not work efficiently in a Wide Area Network (WAN) environment. It was designed directly for local area network authentication and not for globally distributed file sharing authentication. It provides for no encryption of data or electronic mail

transmission. Also, it does not provide for password validation of individual work stations.

J. THE FICUS DISTRIBUTED FILE SYSTEM

1. System Design Goals

The Ficus Distributed File System⁶⁵ is a file replicating system for the UNIX environment intended for use in a very large, wide area network. It employs an optimistic "one copy availability" model where conflicting updates to a file systems directory information are automatically reconciled and conflicting file modifications are reliably detected and reported. The system architecture is based on stackable layers which permit a high degree of modularity and extensibility of file system services.⁶⁶

For each user visible file, there are one or more replicas of the file stored and maintained transparently to the user. Ficus, upon a user request, will automatically select a replica to service the request and propagate all modifications to the replicas. Ficus will also maintain a track on which replicas need updating if they are initially unavailable, for automatic updating when they do become available.

⁶⁵Ficus was developed under ARPA contract No. F29601-87c-0072.

⁶⁶Popeck, G., et al, "Replication in Ficus Distributed File Systems", Department of Computer Science, UCLA.

2. Theory of Operation

Ficus is a distributed file system designed to run on UNIX network systems ranging from portable units and workstations to large file servers.

Ficus provides for volume replication. A ficus volume is a collection of files and directories which are managed together and form a subtree of the UNIX namespace. Each logical ficus volume is represented by a set of volume replicas which form a collection of "containers" for file replicas. Files and directories within the logical volume are replicated in one or more of the volume replicas. Each individual volume replica is normally stored in one UNIX disk partition.

Ficus allows for volume location information by fragmenting the information needed to locate the volume through a specialized pointer in the volume called a graft point. A graft point maps a set of volume file replicas to a host. The host maintains a private table which maps volume replicas to a specific storage device. This type of configuration allows specific volume replica access information to be stored and accessed where it is most needed, rather than using a large, monolithic mapping mount table at each site.

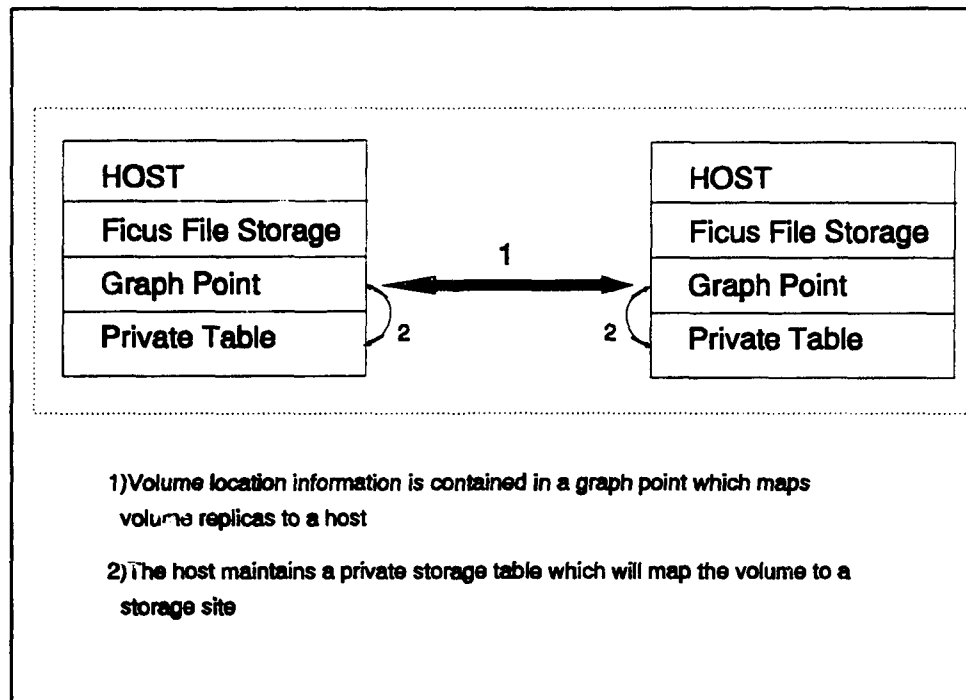


Figure 12. Logical Ficus Volume Set or Container

A Ficus graft point may be replicated like any file or directory in a volume. Volume replicas may be moved, created or destroyed as long as the target volume replica and the graft point replica are available in the UNIX partition, which enforces the concept of "one copy availability."

Ficus supports a very high availability for both read and write and allows for uncoordinated updating when at least one of the replicas of the file are available.⁶⁷ Asynchronous update propagation is provided to accessible copies on a best effort basis but is not relied upon for correct disseminations. Ficus provides for "periodic reconciliation"

⁶⁷Reiher, P., et al, "TRUFFLES - A Secure Service for Widespread File Sharing", Proceedings: Workshop on Networks and Distributed System Security, February 1993, pg 106.

which ensures that, over time, all replicas will be accessed and converged to a common state. In other words, the essential elements of this optimistic replica consistency strategy lie within the Ficus reconciliation algorithms which ensure eventual mutual consistency of all file, directory, and graft point replicas.

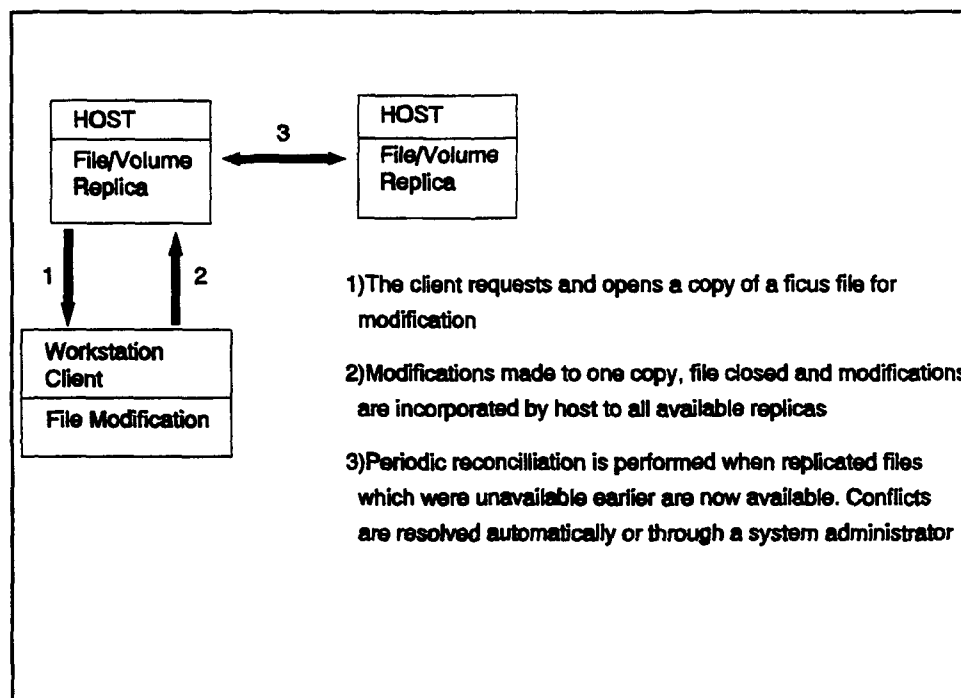


Figure 13. Ficus File Modifications

Because of the asynchronous nature of the propagation strategy and the optimistic one copy availability policy, update propagations of two different replicated ficus files can come into conflict. A conflict occurs when two or more replicas receive updates without successfully propagating their updates to other replicas. Generally, conflicts are automatically detected and reconciled, but those conflicts

which the system cannot reconcile are automatically brought to the attention of the owning user for resolution.

The replication service of Ficus is packaged so that it may be inserted above the base UNIX file system on any machine running a stackable file interface. This stackable layer approach to file system design permits adding functionality to an existing system by writing additional code, but not changing any of the existing code in the operating system kernel.⁶⁸

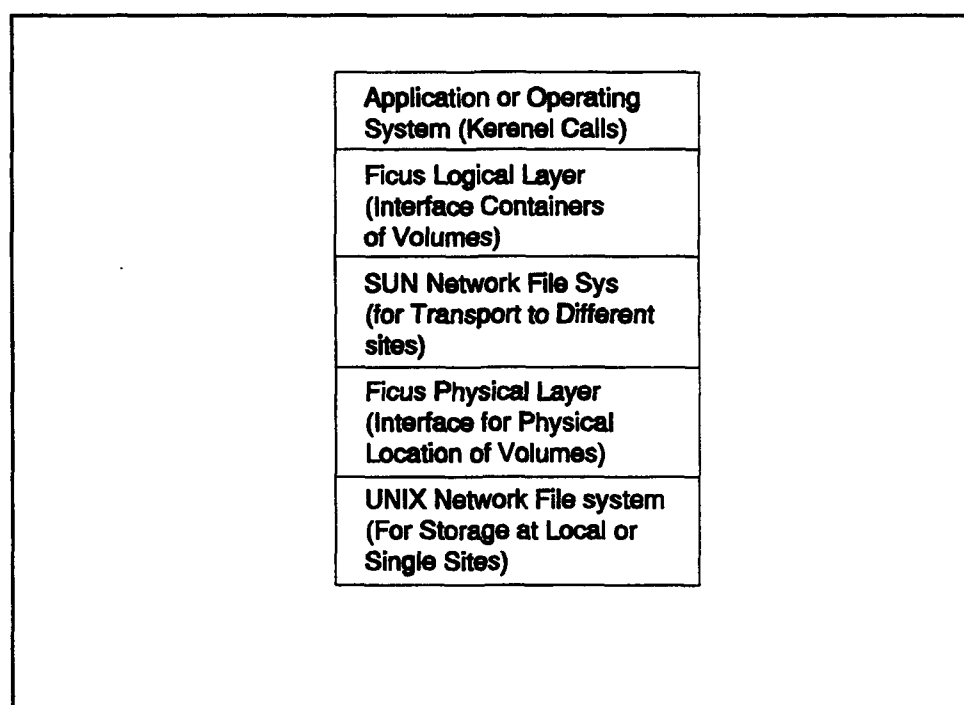


Figure 14. Ficus Stackable Layer Interface

Ficus consists of two layers; a physical layer which supports those operations involving a single replica of a file, and a logical layer which supports operations which

⁶⁸Guy, R.G., et al, "Implementation of the Ficus Replicated File System", USENIX, June 1990.

involve all the replicas of a file. The UNIX Network File System (UFS) is used in the transport layer to move Ficus requests from one site to another.

3. Authentication, Security and File Sharing Issues

The main advantage of the stackable file layers used in Ficus is that other UNIX services can be used in conjunction with this system. Services can be added or deleted by inserting the additional system function into the appropriate layer. Ficus also supports file encryption and compression of files and directories.⁶⁹

K. COMMON CHARACTERISTICS OF REVIEWED DISTRIBUTED FILE SYSTEMS

As the preceding literature review has shown there are many distributed file system technologies, each with its own area of concentration depending on the goal of the particular design. Most systems work on a peer to peer or a client/server relationship. File replication occurs either at the client/workstation or host/server level. Most systems utilize a comparison of cached file timestamps for file consistency. Availability is incorporated through the use of several copies located in multiple servers or multiple access rights (file locking for modifications and concurrency control or read-only access) to specifically connected files. Each system has the

⁶⁹Reiher, P., et al, "Truffles - A Secure Service for Widespread File Sharing", Proceedings: Workshop on Networks and Distributed System Security, February 1993, pg 107.

basic goal of distributing, sharing, and replicating files or directories as well as maintaining file availability and consistency.

Additional concerns incorporating these systems into a DoD organization were their abilities to implement data security measures. Security, usually an afterthought of most civilian oriented applications, is a primary DoD functional requirement. Information flow in the DoD, the private sector, and other Federal agencies has become increasingly sophisticated. Any security breach within an organization can have catastrophic effects on the function of the organization and its environment as a whole. For example, anecdotal evidence suggests, that if Bank of America's computer systems were to be down for 2 days the bank would fold and the economy of California as well as the United States may be crippled.

In the preceding sections, issues discussed were identified as the more popular industry standards for security and encryption. The Digital Encryption Standard (DES), RSA, and the Skipjack algorithm are at the forefront of established encryption technology. Firewall systems and technologies for authentication and access control, such as the Kerberos ticket granting system, are becoming popular as security enhancements to established networks.

One objective of the Corporate Information Management initiative within the DoD is to combine the best of all currently available systems into one architecture which would

ensure the widespread distribution of files and directories while simultaneously ensuring secure transmission and integrity of the information. The goal would be to incorporate one system DoD wide which would provide for mutual user authentication, user validation, non-repudiation through digital signature and password technologies. The DoD system should also support distributed file functions such as high availability, rapid replication, and file consistency and concurrency. Finally, the system should also employ root level transmission security features which provide for peer to peer data encryption/decryption in a user transparent format.

Many of these desired features can be found in the TRUFFLES implementation which is a hybrid of the Ficus distributed file system with an augmentation of privacy enhanced mail (TIS/PEM).

IV. TRUFFLES

This chapter discusses the characteristics of one wide area distributed file system, the Trusted Ficus File System (TRUFFLES), in relation to the final research question: What secure distributed file system currently available can best fit the needs of the DoD? How can it be implemented into an existing network system; specifically, into the network at the Naval Postgraduate School? Discussed are its associated architecture and a general theory of operation. Also discussed are issues concerning how TRUFFLES proposes to overcome the issues of authentication, security, and file sharing methodologies in a manner different than previously discussed distributed file systems or services.

A. DISTRIBUTED FILE SYSTEMS AND TRUFFLES

As we approach nation-wide integration of computer systems, it is clear that file replication will play a key role, both to improve data availability in the face of failures, and to improve performance by locating data near where it used to be.⁷⁰

As stated earlier, the distribution, replication, and sharing of files between users in different administrative

⁷⁰Guy, R.G., et al, "Implementation of the Ficus Replicated File System", USENIX, June 1990.

domains is required within the DoD. With the weight of paper literally weighing down ships, an example of this is the USS Ticonderoga (CG-47) class Aegis cruisers which carries over 72,000 pounds of paper. This translates into almost 1,800 cubic feet of space. Bringing hard copy data into an operational arena is often infeasible, therefore, the digital transfer of information is a viable alternative to paper oriented systems. As a result, many units have initiated a move from paper publications to digital technology, such as CD-ROM.

Paper publication modifications are often complicated, confusing, and slow to implement, resulting in several versions of outdated "current copies" maintained in the users libraries.

With distributed file systems, an important document can be updated at one site and then propagated or replicated to all other sites where the document exists, almost immediately and with little need to interface with the local user. When the local user retrieves the document, they will be notified of any pending modifications and take appropriate action, either replace with the new update, do not replace, or keep both.

The question of information security occurs when discussing transferring data over an untrusted, or common media. One solution is to encrypt the data at end user points. Another reoccurring question is in authenticating and

validating the end user for data transmission. A solution to this problem is to incorporate a mutual authentication or certification mechanism, a "handshake", prior to encrypted data transfer.

One implementation that has been successful in incorporating the previously described features is TRUFFLES (TRUsted Ficus FiLE System). TRUFFLES is a UNIX supported combination of Privacy Enhanced Mail (PEM) and an encryption enhanced version of the Ficus distributed file system. The PEM provides the "handshake" among users and the Ficus distributes the files securely. TRUFFLES allows for file updates as long as at least one replica of the file is available somewhere in the TRUFFLES shared net.

The unique aspect of TRUFFLES is that wide area security enhancements have been incorporated into the overall design. The underlying premise of most secure wide area file sharing system models are to couple a generic privacy enhanced mail implementation to any generic distributed file system for the secure transfer of data.

B. TRUFFLES

TRUFFLES uses Trusted Information System's PEM (TIS/PEM) which authenticates and encrypts electronic mail (using RSA encryption) to establish the "handshake" between users, to determine that a relationship is desired, and then to authenticate the users in the relationship. Electronic mail

also handles the issue of reception failure at destination nodes by reestablishing contact when a site is functioning again. TIS/PEM also establishes encryption keys for the file sharing relationship. After the keys have been established the trusted Ficus⁷¹ portion of the system takes over and handles the file transfer protocols.

After the handshake has been established between the users Ficus will build the "volume". A volume is similar to the concept of a UNIX file system. It consists of a connected tree-like structure of directories and files all stored on a single physical device. Any files that are to be replicated must be collected into a volume or set of volumes.⁷² Ficus ensures the volume of one user will automatically produce an identical volume for the other user or users.

This "free sharing" releases control of the system from the system administrators in areas concerning which of their sites' files are to be shared, by whom, and with whom. This is a problem that has been addressed; in other words, questioning what policies should be established concerning the sharing of trusted, sensitive information. TRUFFLES refers to a policy module during the handshake to determine if the system will

⁷¹Trusted Ficus is the Ficus distributed file system incorporating the DES encryption algorithm.

⁷²Reiher, Peter, Thomas Page, Gerald Popek, Jeff Cook, "Truffles - A Secure Service for Widespread File Sharing" Proceedings: Workshop on Networks and Distributed System Security, February 1993, pg 103.

permit the relationship. The system administrator will set up the provided policy module⁷³. This procedure occurs at both ends of the relationship.

C. FILES/VOLUMES

In each system's hierarchy there is a partition between TRUFFLES files and non-TRUFFLES files. The files stored under the normal UNIX file system are unable to access TRUFFLES files. Only TRUFFLES volumes (collection of directories and files) can be shared with other TRUFFLES volumes. This is a desired aspect to the positive security implications of TRUFFLES.

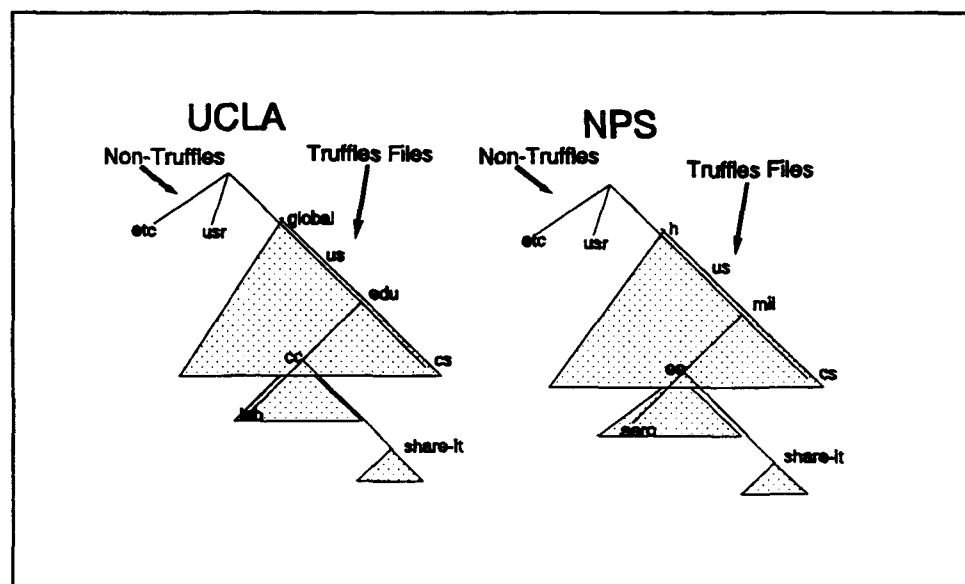


Figure 15. File Hierarchies Using Truffles

⁷³A policy module is a separate program which establishes controls on the system at the discretion of the system administrator; for example, timed access control, and copy permissions.

Within the TRUFFLES portion of the file hierarchy shown in Figure 15, the volume sharing limitations are illustrated by the shaded triangles. The two sites do not necessarily share common namespaces; for example, the root at UCLA is "global" while the root at Naval Postgraduate School (NPS) is "h". However, some portions of the namespace are shared. The TRUFFLES volume at UCLA's **global/us/edu/cc/share-it** has a replica at Naval Postgraduate School **h/us/mil/ee/share-it**. Despite the two replicas being stored at different places in the hierarchies, TRUFFLES keeps all files in the two replicas consistent. Sites are permitted to completely share identical TRUFFLES namespaces.⁷⁴ The portion of the namespace not shared between UCLA and NPS is not accessible to remote sites through TRUFFLES. For example, users at NPS are not able to see UCLA's volume **global/us/edu/cc/lab**, and cannot use TRUFFLES to access it in any way. In fact, the volume **share-it**, as shown in Figure 15, is the only group of files that would be jointly accessible to both UCLA and NPS.

⁷⁴Rieher, Peter, Thomas Page, Jr, Gerald Popek, Jeff Cook, "Truffles - A Secure Service For Widespread File Sharing" Proceedings: Workshop on Networks and Distributed System Security, February 1993, pg 103.

D. TRUFFLES ARCHITECTURE

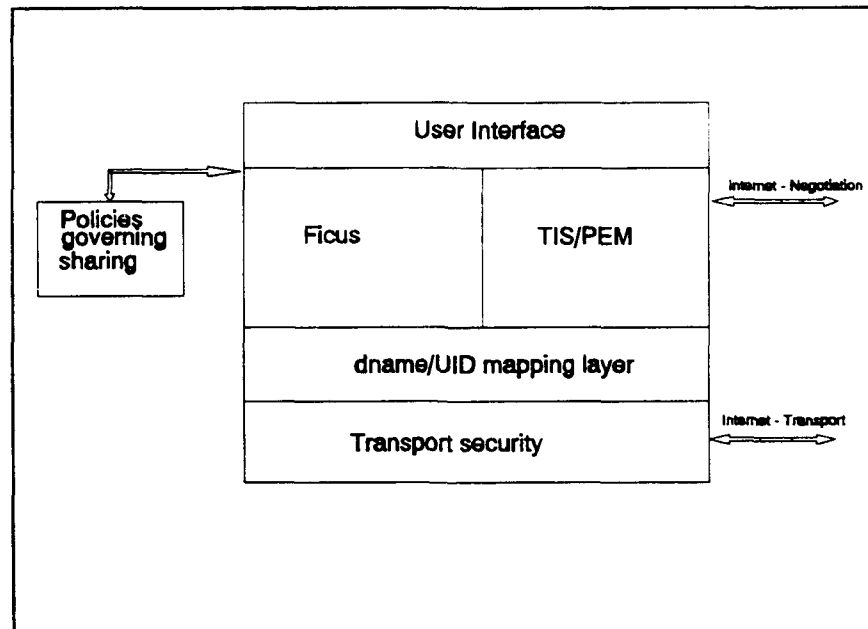


Figure 16. TRUFFLES Architecture

The User Interface interacts with either the Ficus and/or the TIS/PEM portions of TRUFFLES. It also coacts with the policy module for local system administrator controls, such as sharing. When off site transport is required the system calls go through TIS/PEM. TIS/PEM securely transports requests to the TIS/PEM installation at the other site, which then either calls the appropriate Ficus routine or passes the request up through the user interface.⁷⁵ During normal operations, TRUFFLES requests pass through Ficus. Before going off site, the requests pass through a user identifier (UID) mapping

⁷⁵Reiher, Peter, Thomas Page, Gerald Popek, Jeff Cook, "TRUFFLES - A Secure Service for Widespread File Sharing" Proceedings: Workshop on Networks and Distributed System Security, February 1993, pg 105.

layer for the information on the destination's domain. The TRUFFLES requests also pass through transport security.

E. TIS/PEM ARCHITECTURE

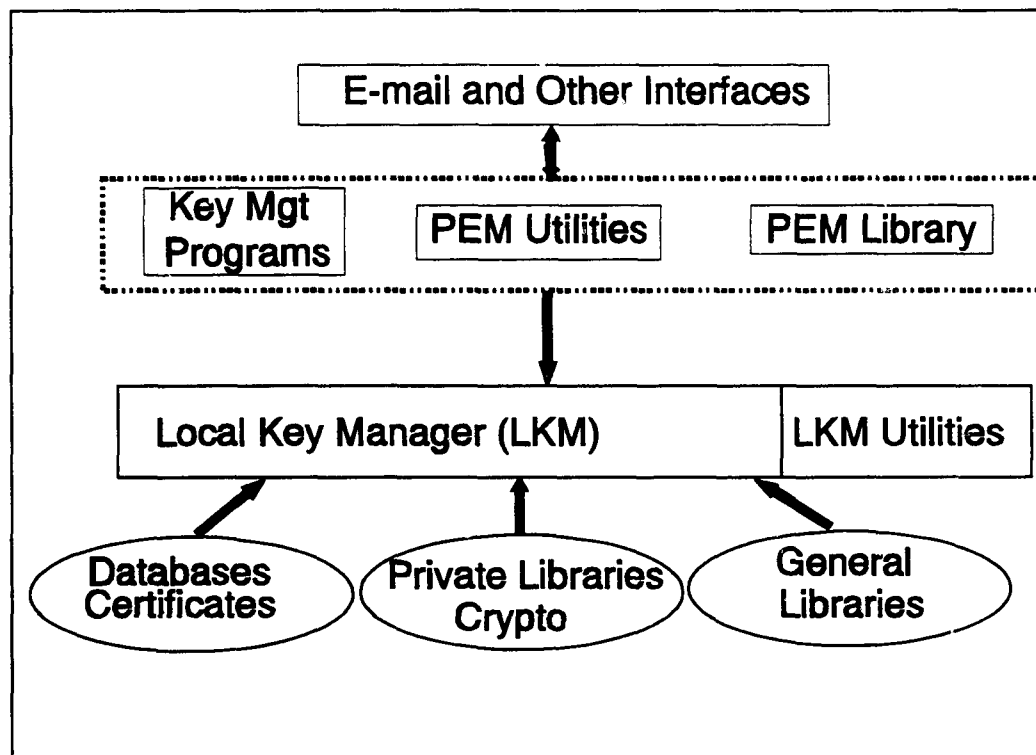


Figure 17. TIS/PEM Interface Architecture

Electronic Mail or other interface services enter the PEM system through a gateway in the PEM library. The PEM library, PEM utilities, and Key Management programs all communicate with an individual designated as the Local Key Manager (LKM). The LKM maintains a local database for certificates and associated private keys. The private libraries, which include the cryptological libraries as well, and general libraries all compose the information repository for the LKM. These three

library modules constantly interact with each other to provide for a smoothly functional implementation.

F. PRACTICALITY OF INSTALLING TRUFFLES AT NPS

The TRUFFLES system is, abstractly, the Ficus distributed file system combined with a Digital Encryption Standard (DES) algorithm and implemented in the kernel of SunOS version 4.1 which, as of this writing, is about 3 years old. Due to the 4.1 operating system kernelization of TRUFFLES, it was deemed impractical to install the TRUFFLES system at the Naval Postgraduate School at the present time. Future upgrades of TRUFFLES will be incorporated into more recent UNIX operating system versions and then the implementation at NPS may be more practical. The specific difficulties of implementation revolve around installing an old operating system on the NPS system which has had at least 5 upgrades since SunOs 4.1. A fiscal concern also plays a role. For the TRUFFLES development team to build the implementation they needed permission to work in the UNIX operating system kernel. This cost the development team \$20,000.00 in 1991. The cost to upgrade to a modern operating system can be expected to higher now.

G. TIS/PEM INSTALLATION AT NPS

In this section, the issues involving the implementation of the installation of Trusted Information Systems Privacy Enhanced Mail (TIS/PEM) are discussed. To acquire TIS/PEM code see APPENDIX C.

The installation of TIS/PEM reference implementation at NPS was performed in July 1994. The code and all associated documentation was downloaded in a compressed format via the UNIX File Transfer Protocol (FTP) from TIS's file transfer site. After the file was uncompressed, the compilable 'C' code and documentation was available and ready for direct installation into the NPS UNIX network system.

TIS/PEM works with most any mail user agent but was designed to be integrated with the Rand MH message handling system (also XMH, and MH under Emacs). In order to be of use with other mail user agents TIS/PEM provides two filters which allows for a quick integration of TIS/PEM into the UNIX environment. These filter modules act as a front end screening interface for different mail handling user agent applications. No attempt was made or deemed necessary to test the filter installation with other available message handling installations. The NPS installation used the RAND MH message handler for which TIS/PEM was originally configured.

The files received for the installation of the TIS/PEM were the following:

1. Numerous documentation and licensing files.
2. mh6.7 - This is MH Version 6.7 with the TIS/PEM integrated.
3. rsaref - This file contains a copy of the RSAREF distribution (RSA's algorithm) and contains all of the cryptographic libraries required for TIS/PEM to operate.
4. pem - This directory contains the complete PEM source distribution; it contains the data dictionaries, manuals, bin directory, and associated commands and calls for PEM required for the UNIX system.

The actual installation TIS/PEM involved first building the RSAREF libraries and files. This process was fairly straightforward and is supported well in the downloaded documentation. Upon successful completion of the RSAREF library construction, the TIS/PEM directory, libraries, and files were built. At this point the system could have been configured, with filters, to use almost any mail user agent. Since the intention was to use the provided Rand MH tool, the provided file mh6.7 was utilized for this purpose.

The entire installation of TIS/PEM was readily accomplished with the most difficulty encountered in the building of the message handler with the TIS/PEM. Even though this implementation was built into the mail user agent, some additional system configuration was necessary. This mostly involved configuring the system for NPS network user requirements. Of the 24 files that could have been configured from the default values, only 4 required minor modification for the system.

The specific files that were reconfigured were the following:

1. "ca_dname": This was the distinguished user name and was subsequently changed from a single user mode to a multi-user mode.
2. "DATADIR" : This was a path name change to identify where the local database was located.
3. "u_access": This was a very important file that would either allow users to make their own certificates or have the system administrator issue the certificates. The method chosen to implement was to allow the individual users make their own certificates; the reason being to reduce the administrative burden of certificate management on the current workload of the system administrator.
4. "pem_domain": This is a file which was an administrative domain configuration and was modified for the NPS systems configuration.

After a successful installation of PEM, there was some difficulty in configuring the system for user certificate issuance. The system administrator (also the Certificate Administrator) was able to create his own certificate as the users guide directed. However, when any other user attempted to create a new certificate, the system would first check to discern if the user held a current certificate and found that the user already held a current certificate. Further investigation into this situation revealed that the certificate each user had was the Certificate Administrator's certificate. The correction to this problem turned out to be a simple oversight on the installation, specifically omitting the "commenting out" marker (similar to removing the REM

command in DOS batch files) of the "u_access" file that was configured to allow users to have their own certificates. The system then performed as advertised. Also, extraneous information was removed from the certificate to allow for more efficient use.

V. CONCLUSIONS

This research has defined a secure distributed file system as a mechanism which allows for sharing and replication of electronic files utilizing end to end encryption across a wide area network. We view it as a logical set of functional components and not merely as a set of connected computer systems.

The qualities and technologies which we feel a secure distributed file system should have are consistently high data availability, system reliability, replication concurrency, file reconciliation, authentication and validation of users, seamless user interface, security capabilities, and open system integration to comply with current standards.

We have reviewed eleven distributed systems or services with each having a particular design goal. Each system had the basic goal of distributing, sharing, and replicating files and directories as well as maintaining file availability, consistency, and concurrency. Most systems incorporated a client/server architecture for their particular mode of operation. File replication occurred at either the client/workstation or the server/host level. File timestamps were predominantly used for file reconciliation. Also, most systems implemented access control through file locking mechanisms to ensure file concurrency.

After reviewing all of the systems and associated services, we have concluded that the best system to provide all of the desired qualities of a secure distributed file system is the TRUsted Ficus FiLE System (TRUFFLES) implementation. The reason for this is that TRUFFLES incorporates all of the aforementioned features as well as the security enhancements designed into the overall configuration and not merely added after the system was developed. Also of importance is the one copy availability feature which allows for complete data replication to be promulgated from one single copy anywhere within the TRUFFLES network. A final, yet critically important feature is the ability of system administrators to configure a provided policy module to seamlessly control the local implementation while still providing integration throughout the entire system.

The Department of Defense applicability and potential usability is outstanding. One of the main uses will be the secure electronic transfer of unclassified sensitive (level 2) or unclassified non-sensitive (level 3) data files. For example, a large sensitive database (Consolidated Ship's Listing (COSAL)) could be promulgated or updated instantaneously. Large groups of data (volumes) could also be transferred with the same ease as single files. An example of this would be the transfer of whole series of publications to an aviation squadron or a ship upon establishment.

Although there are many justifiable concerns surrounding the transmission of secure or sensitive information over common or untrusted media, it must be understood that most of the information transmitted and utilized within the DoD is of Level 2 and 3 security (sensitive/unclassified and unclassified). With this level of security, DES/RSA encryption schemes ensure adequate security in an already established "user friendly" format.

Certification management and the distribution of public and private keys for asymmetrical encryption and digital signature generation are technologically feasible and require only proper administrative controls to be established.

The ideal wide area file distribution system would be to incorporate the replication, availability and security features of TRUFFLES into a DoD wide network, such as the Digital Defense Network, and to provide for local system administration and security through a firewall system, such as Kerberos from project Athena at M.I.T. The firewall system would provide for local system protection against intrusion from hackers or viruses, limit a security breach to a local area if the system is compromised, provide a locally generated audit trail, and provide of a network independent method for modification and maintenance of the local system.

The DoD should also develop regional key distribution centers for key generation and certificate management. Regional centers will provide the same benefits as a single

centralized certificate management facility but limit the scope of the damage and ease of recovery should a security breach occur and key generation be compromised.

The implementation of TIS/PEM at the Naval Postgraduate School proceeded in accordance with the documentation provided by Trusted Information System; however one difficulty was encountered but easily corrected. The actual utilization of TIS/PEM can be accomplished with little formal training. The users manual is clear and concise, and system commands are limited to a minimum, users will become easily accustomed to its operations and usability for secure e-mail. A sample e-mail document with PEM encryption is in APPENDIX B.

The TRUFFLES system is implemented in the kernel of SunOS version 4.1. We deemed it impractical to install at the Naval Postgraduate School because of the antiquity of the operating system. The specific difficulties of implementation revolved around the installation of an old operating system on the NPS system which has had at least 5 upgrades since SunOs 4.1. Future upgrades of TRUFFLES will be incorporated into more recent UNIX operating system versions and then the implementation at NPS may be more practical. With this research it is our intention to notify Computer Center management of the existence of TRUFFLES, and to plan for implementation as soon as the upgrade to a modern operating system is made available.

TRUFFLES architecture and the TIS/PEM reference implementation work in concert to provide the secure distribution of files and directories. The inclusion of a firewall at the network server further enhances the security of the local system. Security from intrusion is a popular issue among system administrators and information professionals. The combination of TRUFFLES, TIS/PEM, and a firewall system will provide the best combination for an integrated and defendable computer network.

The development of distributed file systems architecture in the networked environment has opened up interoperability among systems to a more complex level. Simple file transfer, a basic element of networks, has been elevated to a seamless and secure transfer of files without requiring user or system administrator input. The ability to send large volumes of secure information, nearly instantaneously, has been achieved, and the systems are available for DoD implementation.

Further indepth study is also available concerning the suitability of distributed file systems with video multicasts. Preliminary investigation has revealed that audio is being encrypted, but video images have not been extensively encrypted. A question to answer is can encryption of video be compatible with compression algorithms and is the resulting overhead extensive.

There are many other issues for further study; specifically, naming specifications between heterogeneous

systems, secure transmission data overhead, administrator workload, security certificate issuance, interaction with non-TRUFFLES systems, portability to other hardware and software platforms, and file/volume reconciliation techniques.

APPENDIX A: THE OSI MODEL

The Open System Interconnection (OSI) model is a seven layer protocol model whose intent was to coordinate the design of computer communications architecture and provide a framework for developing protocol standards. Briefly, it is composed of the following hierarchical functions:

- Layer 1: Physical Link: Concerns the transmission of a structured bit stream over a physical medium (cables/connections/couplers).
- Layer 2: Data Link: Provides for reliable transfer of information across the physical link by sending blocks of data (frames) with synchronization, error, and flow controls.
- Layer 3: Networks: Provides for proceeding layers with independence from data transmission and switching systems used in establishing, maintaining, and terminating connections.
- Layer 4: Transport: Provides for reliable, transparent data transmission for end to end communication, error recovery and flow control.
- Layer 5: Session: Provides for the control structure for communications between applications and manages the session between applications.
- Layer 6: Presentation: Provides for independence of the application with regards to the syntax of the data.
- Layer 7: Application: Provides for the design of user interfaces for constructed system incorporating the OSI model and distributed information services.⁷⁶

⁷⁶Malamud, Carl, STACKS, Interoperability in Today's Computer Networks, New Jersey, Prentice Hall, 1992, pg 4.

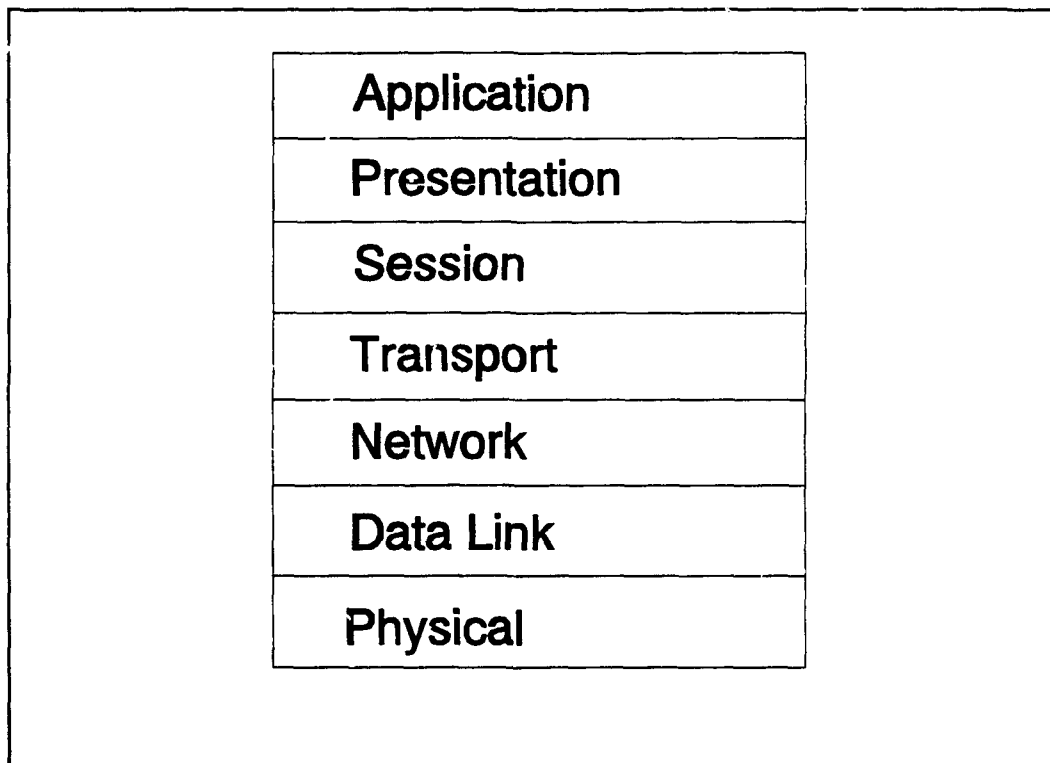


Figure 18. OSI Model

File Transfer, Access and Management (FTAM) is an OSI application layer service which provides for access to foreign file storage systems. Data Access Protocol (DAP) resides in the presentation layer and is used in the Digital Network Architecture to provide for functionality in exchanging data between two network nodes. File Transfer Protocol (FTP) is an Internet standard application layer protocol designed for transferring files from one computer to another. Simple Mail Transfer Protocol (SMTP) is an Internet standard protocol used for transferring electronic mail messages from one computer to another. It specifies how E-mail will interact and the control format messages exchanged to transfer mail.

Most of these protocols are collectively named under Transmission Control Protocol/Internet Protocol (TCP/IP) which refers to a group of application and transport protocols used within the Internet. These include FTP and SMTP.

APPENDIX B: ENCRYPTED PEM MESSAGE

Example of an Encapsulated ENCRYPTED PEM Message

-----BEGIN PRIVACY-ENHANCED MESSAGE-----

Proc-Type: 4, ENCRYPTED

Content-Domain: RFC822

DEK-Info: DES-CBC, BFF968AA74691AC1

Originator-Certificate:

MIIB1TCCASccAWUwDQYJKoZIhvcNAQECBQAwUTELMAkGA1UEBhMCVVMx
IDAeBgNVBAoTF1JTQSBYXRBhIFN1Y3VyaXR5LCBjbmuMQ8wDQYDVQQQL
EwZCZXRhIDExDzANBgNVBAsTBk5PVEFSWTAeFw05MTA5MDQxODM4MTda
Fw05MzA5MDMxODM4MTZaMEUxCzAJBgNVBAYTA1VTMSAwHgYDVQQKEXdS
U0EgRGF0YSBTZWNIcm10eSwgSW5jLjEUMBIGA1UEAxMLVGZvdCBVc2Vy
IDEwWTAKBgRVCAEBAGICAANLADBIakeAwHZH17i+yJcqDtjJCowzTdBJ
rdAiLAnSC+CnnjOJELyuQiBgkGrgIh3j8/x0fM+YrsyFlu3FLZPVtzn
dhYFJQIDAQABMA0GCSqGSIb3DQEBAgUAA1kACKr0PqphJYw1j+YptcIq
iWlFPuN5jJ79Khfg7ASFxskyYkEMjRNZV/HZDZQEhtVaU7Jxfzs2wfx5b
yMp2X3U/5XUXGx7qusDgHQGs7Jk9W8CW1fuSWUGN4w==

Key-Info: RSA,

I3rRIGXUGWAF8js5wCzRTkdhO34PTHdRZY9Tuvvm03M+NM7fx6qc5
udixps2Lng0+wGrtiUm/ovtKdinz6ZQ/aQ==

Issuer-Certificate:

MIIB3DCCAUGCAQowDQYJKoZIhvcNAQECBQAwTzELMAkGA1UEBhMCVVMx
IDAeBgNVBAoTF1JTQSBYXRBhIFN1Y3VyaXR5LCBjbmuMQ8wDQYDVQQQL
EwZCZXRhIDExDTALBgNVBAsTBFRMQ0EwHhcNOTeWOTAxMDgwMDAwWhcN
OTIwOTAxMDc1OTU5WjBRMQswCQYDVQQGEwJVUzEgMB4GA1UEChMXU1NB
IERhdGEgU2VjdXJpdHksIEluYy4xDzANBgNVBAsTBk1ldGEgMTEPMA0G
A1UECxMGTK9UQVJZMHAwCgYEVQgBAQICArwDYgAwXwJYCsnp6lQCxYyk
NlODwutF/jMJ3kL+3PjYyHowk+/9rLg6X65B/LD4bJHtO5XWcqAz/7R7
XhjYcm0PcqbzdzoACZtIleTrKrcJiDYOP+DkZ8klgCk7hQHPbIwIDAQAB
MA0GCSqGSIb3DQEBAgUAA38AAICPv4f9Gx/tY4+p+4DB7MV+tKZnvBoy
8zgoMG0xdD2jMZ/3HsyWKWgSF0eH/AJB3qr9zosG47pyMntf3aSy2nBO
7CMxpUWRBcXUpE+xEREZd9++32ofGBIXaialnOgVUn0OzSYgugiQ077n
JLDUj0hQehCizEs5wUJ35a5h

MIC-Info: RSA-MD5, RSA,

UdFJR8u/TIGHfH65ieewe2lOW4tooa3vZCvVNGBZirf/7nrgzWDA
Bz8w9NsXSexvAjRFbHoNPzBuxwmOAFa0HJsZL4yBvhG

Recipient-ID-Asymmetric:

MFEXCzAJBgNVBAYTA1VTMSAwHgYDVQQKEXdSU0EgRGF0YSBTZWNI
cm10eSwgSW5jLjEUMBIGA1UEAxMLVGZvdCBVc2VyZDQYDVQQLEwZO
T1RBULK=, 66

Key-Info: RSA,

O6BS1ww9CTyHPtS3bMLD+L0hejdVX6Qv1HK2ds2sQPEaXhX8EhvVphHY
TjwekdWv7x0Z3Jx2vTAhOYHMcqgCjA==qeWlj/YJ2Uf5ng9yznPbtD0m
YloSwIuV9FRYx+gzY+8iXd/NQrXHfi6/MhPfPF3djIqCJAXvld2xgqQi
mUzoS1a4r7kQQ5c/Iua4LqKeq3ciFzEv/MbZhA==

-----END PRIVACY-ENHANCED MESSAGE-----

Legend for PEM message:⁷⁷

1. Proc-Type: Header field, required for all PEM messages. It identifies the type of processing performed on the transmitted message. The types of PEM messages are ENCRYPTED, MIC-ONLY (Message Integrity Checks), MIC-CLEAR, and CRL (Certificate Revocation Lists).
2. Content-Domain Field: Presently describes the type of content which is represented within a PEM message's text. It carries one string, "RFC822" to indicate processing of RFC-822 mail as defined by the specification and more domains are anticipated.
3. DEK-Info Field: (Data Encryption Keys) Identifies the message text encryption algorithm and mode, and also carries any cryptographic parameters.
4. Originator-ID Field: Identifies a message's originator and provides the originator's Interchange Keys (IK) identification component.
5. Issuer-Certificate Field: Used for asymmetric key management. Contains the certificate of the public component used to sign the certificate carried by the "Originator-Certificate:", for the recipient's use in chaining through the certificates certification path.
6. MIC-Info Field: Used only when asymmetric key management is employed. It identifies the algorithm under which the accompanying MIC is computed and signed, as well as the MIC signed with the originator's private key.
7. Key-Info Field: Provides an IK use indicator, MIC algorithm indicator, a DEK, and a MIC.
8. Recipient-ID Field: Identifies a recipient and provides a recipient's IK identification component.

⁷⁷Linn, J., Request for Comments: 1421, February 1993, pgs 23-30.

APPENDIX C: ACQUIRING TIS/PEM⁷⁸

TIS/PEM, a reference implementation of Privacy Enhanced Mail (PEM) is available for broad use and provided by RSA Data Security, Inc., and Trusted Information Systems, Inc. It is distributed in source code form, with all modules written in C programming language. TIS/PEM runs on most UNIX platforms and is integrated with the following user interfaces: Version 6.7 of the Rand MH Message Handling system, XMH, and MH under Emacs. Filters are provided to work with other message handling systems.

The main cryptography package supplied by RSA Data Security, Inc. is the RSAREF. This package is licensed to only be used with TIS/PEM and only within the United States and Canada.

The source code for TIS/PEM is found in a blind directory and it is a zipped tar file (p-6.1.tar.Z). These are the steps to download the source code and documentation.

- FTP to ftp.tis.com.
- Go to the directory pub/PEM and download the README file.
- Read the file and it will instruct you as to which blind directory the code is located. The directory information

⁷⁸Trusted Information Systems, Inc., TIS/PEM User's Guide, October 1993.

will be good for 10 minutes, and also informs you of some set up required for postscript transformation (docs).

- The file will be found in a path similar to this: pub/PEM/dist/pem-XXXXXX/p-6.1.tar.Z.

Additional information regarding support of TIS/PEM can be found via the user's group

<tispem-users@tis.com>.

To join send e-mail to

<tispem-users-request@tis.com>

and ask to be added to the distribution list.

For technical questions send e-mail to

<tispem-support@tis.com>.

For certificate registration information send e-mail to

<tispca-info@tis.com> or <pca-info@rsa.com>.

Send correspondence to TIS at:

Trusted Information Systems

3060 Washington Road (Rt. 97)

Glenwood, Maryland 21738

REFERENCES

- Arnold, N., UNIX Security: A Practical Tutorial, McGraw-Hill, Inc., 1993.
- Avolio, F.M., Ranum, M.J., "A Network Perimeter with Secure External Access", Trusted Information Systems, Inc, January 25, 1994.
- Cohen, D., "AFS: NFS on Steroids", LAN Technology, March 1993.
- Committee of Government Operations, Sixth Report on DoD Automated Information Systems, Report 101-382, November 16, 1989.
- Denning, D., "The Clipper Chip: A Technical Summary", April, 1993.
- Digital Signature Standard (DRAFT), National Institute of Standards and Technology, February 1, 1993.
- DoD Trusted computer system Evaluation criteria, DoD 5200.28-std, Library No. S225,711, December 1985.
- Guy, R.G., et al, "Implementation of the Ficus Replicated File System", USENIX, June 1990.
- Helsing, Cheryl, "Executive Guide to the Protection of Information Resources", National Institute of Standards and Technology, date unknown.
- Hughes, Larry J., "A Brief Overview of Kerberos Authentication", Indiana University, University Computing Services Network Applications Group, October 1993.
- Kellner, Mark, "Data Power", Government Executive, August 1991.
- Kent, S. "Privacy Enhancement for Internet Electronic Mail: Part II Certificate-based Key Management", Network Working Group RFC 1422, February 1993.
- Koerner, Frank, "System Threats and Vulnerabilities and the Contrary Principle", Computer Humanware International, Elsevier Science Publishers Ltd, 1993.

- Linn, J. "Privacy Enhancement for Internet Electronic Mail: Part I Message Encryption and Authentication Procedures", Network Working Group RFC 1421, February 1993.
- Malamud, C., STACKS, Interoperability in Today's Computer Networks, Prentice Hall, 1992.
- Muftic, Sead, "Security Mechanisms for Computer Networks", Ellis Horwood Limited, 1989.
- Popeck, G., et al, "Replication in Ficus Distributed File Systems", Department of Computer Science, UCLA.
- Ranum, Marcus J., "A Network Firewall", Digital Equipment Corporation, Washington Open Systems Resource Center, Greenbelt, MD., June 12, 1992.
- Ranum, Markus J., "Thinking About Firewalls", Trusted Information Systems, Inc., Glenwood Maryland.
- Reiher, P., T. Page, G. Popek, "TRUFFLES - A Secure Service for Widespread File Sharing", Proceedings: Workshop on Networks and Distributed System Security, February 1993.
- Riordan, M., "RIPEM Users Guide: for RIPEM vers. 1.2", January 1994.
- Russell, D. and Gangemi, G.T., Computer Security Basics, O'Reilly and Associates, Inc, July 1992.
- Satyanarayanan, M., "A Survey Of Distributed File Systems", School of Computer Science, Carnegie Mellon University, Annual Review for Computer Science, 1990.
- Satyanarayanan, M., et al, "Coda: A highly Available File System for a Distributed Workstation Environment", IEEE Transactions on Computers, Vol 39, No 4, April 1990.
- Schneier, Bruce, "Digital Signatures", BYTE magazine, November 1993.
- Stallings, W., Data and Computer Communications, 4th Edition, MacMillian Publishing Company, New York.
- Trusted Information Systems, Inc., TIS/PEM User's Guide, October 1993.

Walker, B., et al, "The LOCUS Distributed Operating System", Operating Systems Review, Volume 17, Number 5.

INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2. Library, Code 52 Naval Postgraduate School Monterey, California 93943-5101	2
3. Roger Stemp, Lecturer, Code CS/Sp Department of Computer Science Naval Postgraduate School Monterey, California 93943-5000	4
4. Kishore Sengupta, Professor, Code SM/Se Department of Systems Management Naval Postgraduate School Monterey, California 93943-5000	2
5. Joe Blau, Computer Center, Code 51 Naval Postgraduate School Monterey, California 93943-5000	1
6. Winslow H. Buxton 2940 Logan Drive Pensacola, Florida 32503	8
7. Tracy M. Conroy 9346 Babauta Road #80 San Diego, California 92129-4905	8